Taylors maturity sampling data visualisation

* Set up a git repository for your analysis.
* Tracking your work with git.
* Make your data tidy.
* Explore the data through visualisation (min 5 plots).
* Develop a final image that tells the data’s story.
* Make a KnitR report with your code and plots.
* Present to your classmates.

# Setting up working environments.

#### Working directory and folders

In my DATASCHOOL working directory, I created a new folders to organise my work.

I have a folder for my raw data, scripts and figures.

Using the drop down in R studio I have also created files, these files are:

* Project in R studio
* Script files in R (I have multiple files for different task for the project)
* Markdown file

The advantage of creating a new project in R studio is that I am prompted to create a git repository

#### Creating a GIT repository.

When I select create a new project in R studio, I go onto create a new directory and new project, here is a small checkbox ‘create git repository’. Once this is selected I can go to R studio terminal to double check that I have repository.

In the terminal $git status

#### Next step is to go to GitHub

log in, and create a new repository.

I selected public, and simply follow the instructions:

‘….or push on existing repository from command line’

copy the code and paste in R studio terminal - done!

This part is setting up a repository on Github which allows my work to be pushed from my local computer to Github.

#### Staging, committing and pushing in R studio

Again R studio makes things easy.

Git pane has a list of my files.

Checkbox allows me to stage the files and then commit.

Adding a comment on the commit window.

At the end of the day, I can push my files to Github using the push icon in the git window.

# Importing data and make Tidy

#### Import data

* Import tidyverse and ggplot2 (note tidyverse already had ggplot2)
* Import dplyer
* Import my data using read\_csv

library(tidyverse)

## -- Attaching packages ------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.0.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.6  
## v tidyr 0.8.1 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

## -- Conflicts ---------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(dplyr)  
  
maturity\_data <- read\_csv("Raw\_data/Taylors\_maturity\_2015\_2016.csv")

## Warning: Missing column names filled in: 'X19' [19], 'X20' [20],  
## 'X21' [21], 'X22' [22], 'X23' [23], 'X24' [24], 'X25' [25], 'X26' [26],  
## 'X27' [27], 'X28' [28], 'X29' [29], 'X30' [30], 'X31' [31]

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## Year = col\_integer(),  
## `sampling event` = col\_integer(),  
## GFV = col\_integer(),  
## Sample = col\_integer(),  
## `GPS sampling pt ID` = col\_integer(),  
## POINT\_X = col\_double(),  
## POINT\_Y = col\_double(),  
## pH = col\_double(),  
## TA = col\_double(),  
## brix = col\_double(),  
## `bunch wt` = col\_double(),  
## `Mean berry wt` = col\_double(),  
## `berries/bunch` = col\_integer(),  
## `Colour per berry Wt` = col\_double(),  
## `Phenolic per berry Wt` = col\_double(),  
## `Colour per berry` = col\_double(),  
## `Phenolic per berry` = col\_double()  
## )

## See spec(...) for full column specifications.

#### Check out what my data looks like

I have a few options here;

* head()
* str()
* View(maturity\_data)
* glimps

#str(maturity\_data)  
#head(maturity\_data)  
#View(maturity\_data)  
glimpse(maturity\_data)

## Observations: 450  
## Variables: 31  
## $ Year <int> 2015, 2015, 2015, 2015, 2015, 2015, 20...  
## $ `sampling event` <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...  
## $ `Sampling date` <chr> "14/01/2015", "14/01/2015", "14/01/201...  
## $ GFV <int> 2414, 2414, 2414, 2414, 2414, 2414, 24...  
## $ Sample <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,...  
## $ `GPS sampling pt ID` <int> 101, 102, 103, 104, 105, 106, 107, 108...  
## $ POINT\_X <dbl> 286331.7, 286382.4, 286425.7, 286473.1...  
## $ POINT\_Y <dbl> 6236797, 6236810, 6236821, 6236833, 62...  
## $ pH <dbl> 2.724, 2.685, 2.789, 2.801, 2.695, 2.7...  
## $ TA <dbl> 16.579, 16.699, 15.729, 15.304, 22.389...  
## $ brix <dbl> 12.1, 12.1, 12.4, 12.4, 10.7, 11.9, 12...  
## $ `bunch wt` <dbl> 121.55, 158.68, 148.42, 143.55, 100.64...  
## $ `Mean berry wt` <dbl> 1.153, 1.139, 1.268, 1.392, 1.023, 0.9...  
## $ `berries/bunch` <int> 105, 139, 117, 103, 98, 99, 107, 106, ...  
## $ `Colour per berry Wt` <dbl> 0.187, 0.204, 0.249, 0.218, 0.109, 0.1...  
## $ `Phenolic per berry Wt` <dbl> 1.156, 1.221, 1.142, 1.111, 1.126, 1.1...  
## $ `Colour per berry` <dbl> 0.214, 0.243, 0.332, 0.293, 0.116, 0.1...  
## $ `Phenolic per berry` <dbl> 1.324, 1.456, 1.523, 1.495, 1.194, 1.1...  
## $ X19 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ X20 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ X21 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ X22 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ X23 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ X24 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ X25 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ X26 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ X27 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ X28 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ X29 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ X30 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ X31 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...

#### Make my data tidy.

I have a number of problem with my data that needs to be fixed:

1. I have columns that need to be removed, ie I have number of columns that contain NA
2. I have column headings with white spaces
3. I don’t have a unique ID column
4. My date column is not represented at a date rather a string. I should consider splitting this column out to day, month, year

Consider creating a new data column that has year as factor.

#### 1. Select data I want to use / remove the data I don’t want.

Reduce my data set from 31 variable to 18.

mat\_tidy1 <- select(maturity\_data,c(1:18))  
#glimpse(mat\_tidy1)

I played around with a number of other option. (I was not convinced it worked It looked like this worked when you run head but when you run str() you get the original list).

#test1\_1 <- select(maturity, -starts\_with("X"))  
#head(test1\_1)

#### 2. Remove the white spaces in heading names.

This can be done one heading at a time using this function.

names(maturity\_data\_clean)[names(maturity\_data\_clean) == ‘sampling event’] <- ‘sampling\_event’

But much better way is to use this function. Which replaces both the white spaces and / with \_.

names(mat\_tidy1) <-str\_replace\_all(names(mat\_tidy1), c(" " = "\_", "/" = "\_" ))  
#glimpse(mat\_tidy1)

*Help I would like to make a new df with the new headings but I can’t get this to work for me*

#### 3. Create a unique ID.

**3a Create a new data column for year as float**

mat\_tidy1 <- mat\_tidy1 %>%  
 mutate(Year\_a = Year)  
  
mat\_tidy1$Year\_a <- as.factor(mat\_tidy1$Year\_a)  
#glimpse(mat\_tidy1)

\*\*3b create a new data coloum that combines Year and GPS\_sampling\_\_pt\_ID\*\*

mat\_tidy3 <- mat\_tidy1 %>%  
 unite(ID, Year, GPS\_sampling\_\_pt\_ID)  
  
#glimpse(mat\_tidy3)

Another option would be: This would keep input data columns.

unite(ID, Year, GPS\_sampling\_\_pt\_ID, sep = “\_“, remove = FALSE)

#### 4. My date column is not represented at a date rather a string.

I should consider splitting this coloumn out to day, month, year.

mat\_tidy4 <- mat\_tidy3 %>%  
 separate(Sampling\_date, c("Day", "Month", "Year"), sep = '/')  
mat\_tidy4

## # A tibble: 450 x 20  
## ID sampling\_event Day Month Year GFV Sample POINT\_X POINT\_Y  
## <chr> <int> <chr> <chr> <chr> <int> <int> <dbl> <dbl>  
## 1 2015~ 1 14 01 2015 2414 1 286332. 6.24e6  
## 2 2015~ 1 14 01 2015 2414 2 286382. 6.24e6  
## 3 2015~ 1 14 01 2015 2414 3 286426. 6.24e6  
## 4 2015~ 1 14 01 2015 2414 4 286473. 6.24e6  
## 5 2015~ 1 14 01 2015 2414 5 286583. 6.24e6  
## 6 2015~ 1 14 01 2015 2414 6 286631. 6.24e6  
## 7 2015~ 1 14 01 2015 2414 7 286667. 6.24e6  
## 8 2015~ 1 14 01 2015 2414 8 286729. 6.24e6  
## 9 2015~ 1 14 01 2015 2414 9 286759. 6.24e6  
## 10 2015~ 1 14 01 2015 2414 10 286779. 6.24e6  
## # ... with 440 more rows, and 11 more variables: pH <dbl>, TA <dbl>,  
## # brix <dbl>, bunch\_wt <dbl>, Mean\_berry\_wt <dbl>, berries\_bunch <int>,  
## # Colour\_per\_berry\_Wt <dbl>, Phenolic\_per\_berry\_Wt <dbl>,  
## # Colour\_per\_berry <dbl>, Phenolic\_per\_berry <dbl>, Year\_a <fct>

#head(mat\_tidy4)

These columns are character I want them to be integers.

mat\_tidy4$Day <- as.integer(mat\_tidy4$Day)  
mat\_tidy4$Month <- as.integer(mat\_tidy4$Month)  
mat\_tidy4$Year <- as.integer(mat\_tidy4$Year)  
#glimpse(mat\_tidy4)

# Plotting data

### Plot grape variable vs time or degree days.

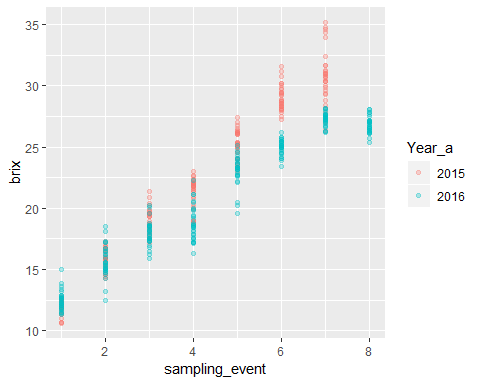
Make an empty plot

#ggplot(data=mat\_tidy4, aes(x = sampling\_event, y = brix))

Now fill my plot with data.

ggplot(data=mat\_tidy4, aes(x = sampling\_event, y = brix, colour = Year\_a))+  
 geom\_point(alpha = 0.3)

## Warning: Removed 1 rows containing missing values (geom\_point).



Each year had a number of sampling events which occurred through the season.

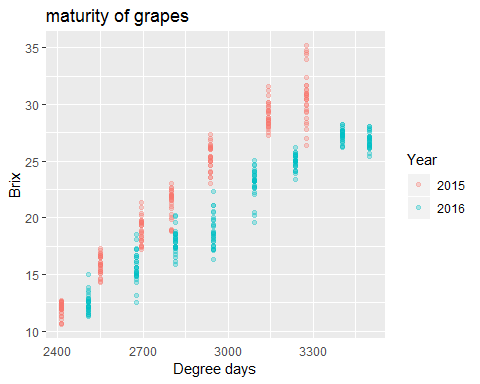
In 2015 we sampled 30 vines on 7 separate events.

In 2016 we sampled 30 vines on 8 separate events.

Plotting by event makes the data points sit ontop of each other. Plotting by date (on the x axis) will mean that there is a big gap between years. I have calculated the growing degree days for each year, based on local weather data, this could be a way of displaying the x axis.

ggplot(data=mat\_tidy4, aes(x = GFV, y = brix, colour = Year\_a))+  
 geom\_point(alpha = 0.3) +  
 labs(x = "Degree days",y = "Brix",title = ("maturity of grapes"),colour = "Year") +  
 theme(axis.text.x=element\_text(), axis.ticks.x = element\_blank())

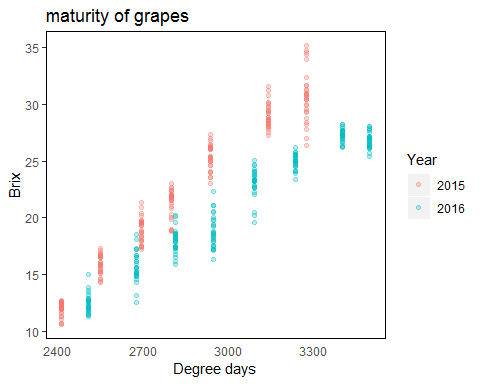
## Warning: Removed 1 rows containing missing values (geom\_point).



Formatting…

ggplot(data=mat\_tidy4, aes(x = GFV, y = brix, colour = Year\_a))+  
 geom\_point(alpha = 0.3) +  
 labs(x = "Degree days",y = "Brix",title = ("maturity of grapes"),colour = "Year") +  
 theme(panel.grid = element\_blank(),  
 axis.text.x=element\_text(),   
 axis.ticks.x = element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

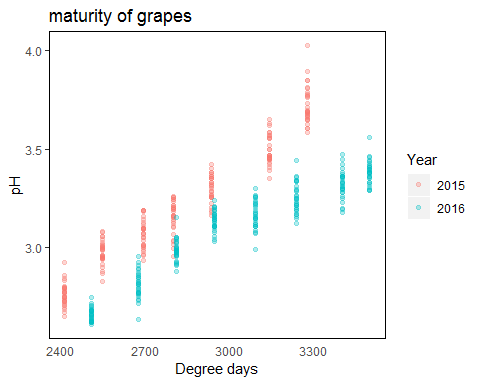
## Warning: Removed 1 rows containing missing values (geom\_point).



## Try all the grape properties…

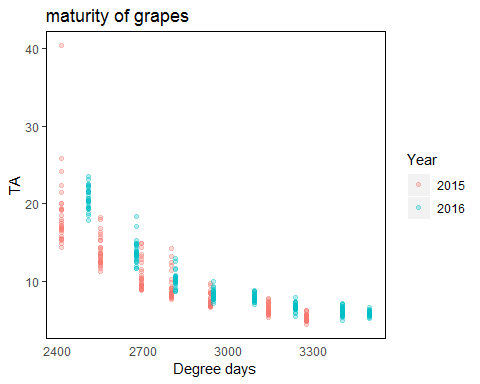
#X = GFV and Y = pH  
ggplot(data=mat\_tidy4, aes(x = GFV, y = pH, colour = Year\_a))+  
 geom\_point(alpha = 0.3) +  
 labs(x = "Degree days",y = "pH",title = ("maturity of grapes"),colour = "Year") +  
 theme(panel.grid = element\_blank(),  
 axis.text.x=element\_text(),   
 axis.ticks.x = element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

## Warning: Removed 1 rows containing missing values (geom\_point).

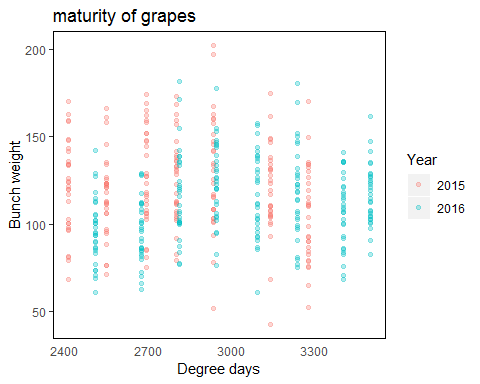


#X = GFV and Y = TA  
ggplot(data=mat\_tidy4, aes(x = GFV, y = TA, colour = Year\_a))+  
 geom\_point(alpha = 0.3) +  
 labs(x = "Degree days",y = "TA",title = ("maturity of grapes"),colour = "Year") +  
 theme(panel.grid = element\_blank(),  
 axis.text.x=element\_text(),   
 axis.ticks.x = element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

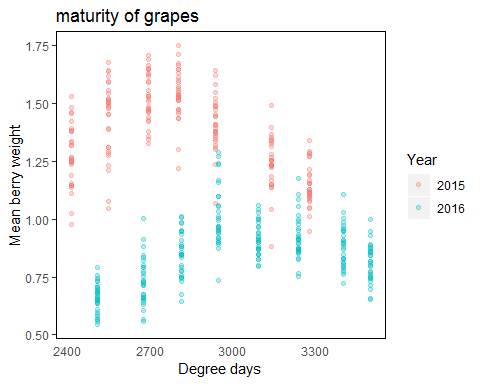
## Warning: Removed 1 rows containing missing values (geom\_point).



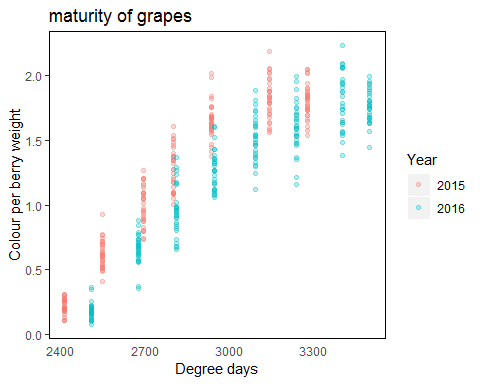
#X = GFV and Y = bunch\_wt  
ggplot(data=mat\_tidy4, aes(x = GFV, y = bunch\_wt, colour = Year\_a))+  
 geom\_point(alpha = 0.3) +  
 labs(x = "Degree days",y = "Bunch weight",title = ("maturity of grapes"),colour = "Year") +  
 theme(panel.grid = element\_blank(),  
 axis.text.x=element\_text(),   
 axis.ticks.x = element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))



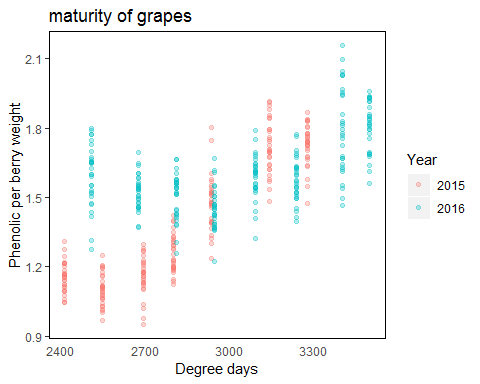
#X = GFV and Y = Mean\_berry\_wt  
ggplot(data=mat\_tidy4, aes(x = GFV, y = Mean\_berry\_wt, colour = Year\_a))+  
 geom\_point(alpha = 0.3) +  
 labs(x = "Degree days",y = "Mean berry weight",title = ("maturity of grapes"),colour = "Year") +  
 theme(panel.grid = element\_blank(),  
 axis.text.x=element\_text(),   
 axis.ticks.x = element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))



#X = GFV and Y = Colour\_per\_berry\_Wt  
ggplot(data=mat\_tidy4, aes(x = GFV, y = Colour\_per\_berry\_Wt, colour = Year\_a))+  
 geom\_point(alpha = 0.3) +  
 labs(x = "Degree days",y = "Colour per berry weight",title = ("maturity of grapes"),colour = "Year") +  
 theme(panel.grid = element\_blank(),  
 axis.text.x=element\_text(),   
 axis.ticks.x = element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))



#X = GFV and Y = Phenolic\_per\_berry\_Wt  
ggplot(data=mat\_tidy4, aes(x = GFV, y = Phenolic\_per\_berry\_Wt, colour = Year\_a))+  
 geom\_point(alpha = 0.3) +  
 labs(x = "Degree days",y = "Phenolic per berry weight",title = ("maturity of grapes"),colour = "Year") +  
 theme(panel.grid = element\_blank(),  
 axis.text.x=element\_text(),   
 axis.ticks.x = element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))



## Summary of grape variable vs degree days.

Most grape properties change as the degree days increase (brix, pH, TA and colour). Handful of grape properties have a high amount of variation and you can’t see change over season (bunch weight, berry weight and phenolics to a degree).

Does the location of the sampling in the vineyard matter?

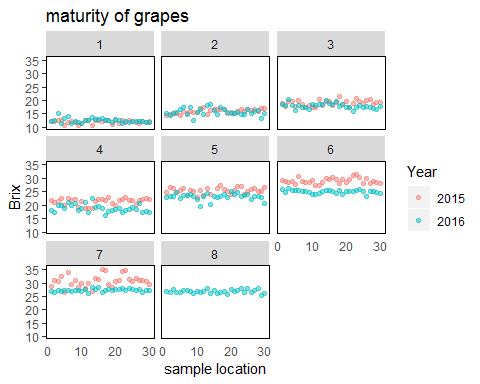
# Do grape properties change depending on loaction.

Plotting variation through spaces vs variation through time. Using brix and bunch weight as example.

## Plot brix for each sample point.

ggplot(data=mat\_tidy4, aes(x = Sample, y = brix, colour = Year\_a))+  
 geom\_point(alpha = 0.5) + facet\_wrap(~ sampling\_event)+  
labs(x = "sample location",y = "Brix",title = ("maturity of grapes"),colour = "Year") +  
 theme(panel.grid = element\_blank(),  
 axis.text.x=element\_text(),   
 axis.ticks.x = element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

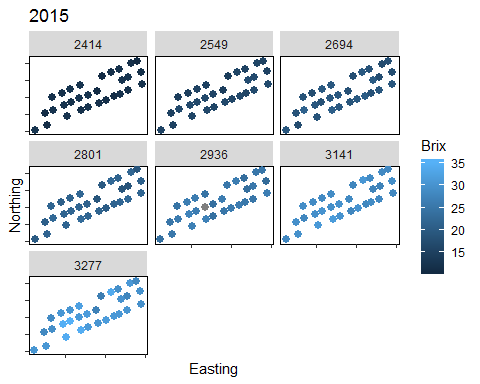
## Warning: Removed 1 rows containing missing values (geom\_point).



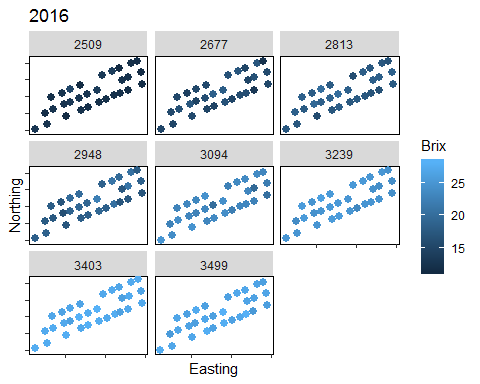
This plot is saying the early in the season differences between vineyard location and year is not as obvious as late in the year. This is quite a broad generalisation.

## Is sample location associated with high brix values?

mat\_tidy4 %>%  
 filter(Year\_a == "2015") %>%  
 ggplot(aes(x = POINT\_X, y = POINT\_Y, colour = brix)) +  
 geom\_point(size = 2.5) +  
 facet\_wrap(~ GFV) +   
 labs(   
 x = "Easting",  
 y = "Northing",  
 title = "2015",  
 colour = "Brix") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_blank(),  
 axis.text.y=element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))



mat\_tidy4 %>%  
 filter(Year\_a == "2016") %>%  
 ggplot(aes(x = POINT\_X, y = POINT\_Y, colour = brix)) +  
 geom\_point(size = 2.5) +  
 facet\_wrap(~ GFV) +   
 labs(   
 x = "Easting",  
 y = "Northing",  
 title = "2016",  
 colour = "Brix") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_blank(),  
 axis.text.y=element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

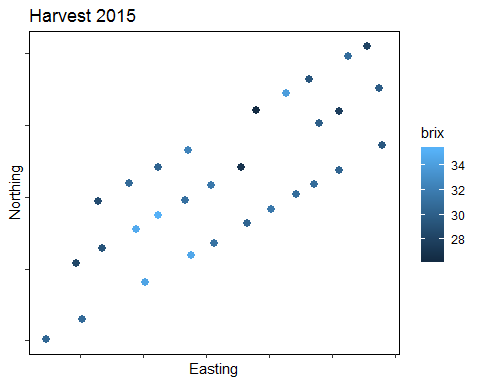


What I really want to see is each pane with its own scale, it is hard to remove the effect of time on these graphs.

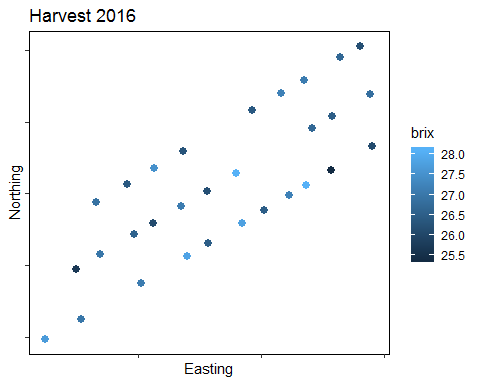
Standardize brix per sampling event.

## Harvest sampling is pretty important.

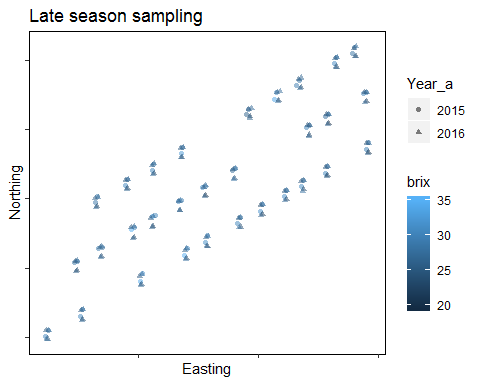
mat\_tidy4 %>%  
 filter(GFV == "3277") %>%  
 ggplot(aes(x = POINT\_X, y = POINT\_Y, colour = brix)) +  
 geom\_point(size = 2.5) +  
 labs(   
 x = "Easting",  
 y = "Northing",  
 title = "Harvest 2015",  
 colour = "brix") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_blank(),  
 axis.text.y=element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))



mat\_tidy4 %>%  
 filter(GFV == "3499") %>%  
 ggplot(aes(x = POINT\_X, y = POINT\_Y, colour = brix)) +  
 geom\_point(size = 2.5) +  
 labs(   
 x = "Easting",  
 y = "Northing",  
 title = "Harvest 2016",  
 colour = "brix") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_blank(),  
 axis.text.y=element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

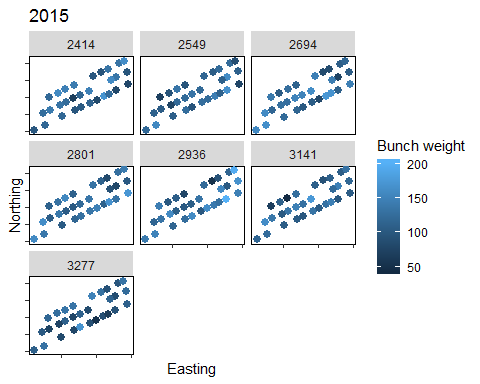


mat\_tidy4 %>%  
 filter(GFV > "3000") %>%  
 ggplot(aes(x = POINT\_X, y = POINT\_Y, colour = brix, shape = Year\_a)) +  
 geom\_point(size = 1.5, alpha = 0.5) +  
 labs(   
 x = "Easting",  
 y = "Northing",  
 title = "Late season sampling",  
 colour = "brix") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_blank(),  
 axis.text.y=element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

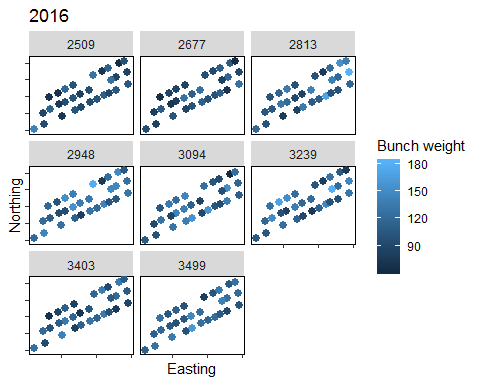


#### Bunch weight

mat\_tidy4 %>%  
 filter(Year\_a == "2015") %>%  
 ggplot(aes(x = POINT\_X, y = POINT\_Y, colour = bunch\_wt)) +  
 geom\_point(size = 2.5) +  
 facet\_wrap(~ GFV) +   
 labs(   
 x = "Easting",  
 y = "Northing",  
 title = "2015",  
 colour = "Bunch weight") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_blank(),  
 axis.text.y=element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))



mat\_tidy4 %>%  
 filter(Year\_a == "2016") %>%  
 ggplot(aes(x = POINT\_X, y = POINT\_Y, colour = bunch\_wt)) +  
 geom\_point(size = 2.5) +  
 facet\_wrap(~ GFV) +   
 labs(   
 x = "Easting",  
 y = "Northing",  
 title = "2016",  
 colour = "Bunch weight") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_blank(),  
 axis.text.y=element\_blank(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))



## Summary of sample location vs grape properties.

Plotting just coordinated vs grape properties is not telling me much.

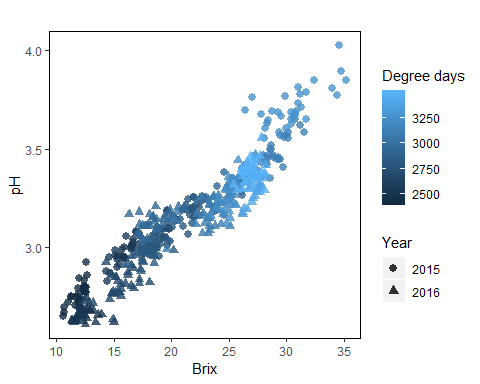
Need to overlay with something else, location in the vineyard is a surrogate for soil condition elevation etc…

I need to add another variable to my data set!

# Plotting grape variable against each other.

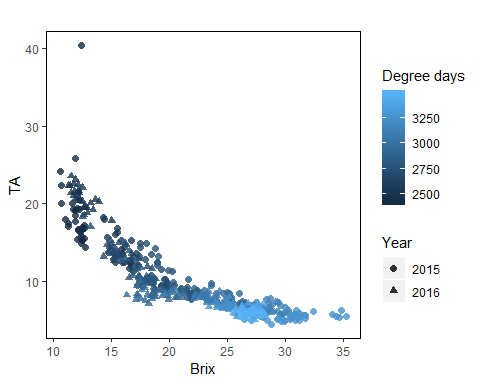
#Brix vs pH  
mat\_tidy4 %>%  
 ggplot(aes(x = brix, y = pH, colour = GFV, shape = Year\_a)) +  
 geom\_point(size = 2.5, alpha =0.8) +  
 labs(   
 x = "Brix",  
 y = "pH",  
 title = "",  
 colour = "Degree days",   
 shape = "Year") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_text(),  
 axis.text.y=element\_text(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

## Warning: Removed 1 rows containing missing values (geom\_point).



#Brix vs TA  
mat\_tidy4 %>%  
 ggplot(aes(x = brix, y = TA, colour = GFV, shape = Year\_a)) +  
 geom\_point(size = 2.0, alpha =0.8) +  
 labs(   
 x = "Brix",  
 y = "TA",  
 title = "",  
 colour = "Degree days",   
 shape = "Year") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_text(),  
 axis.text.y=element\_text(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

## Warning: Removed 1 rows containing missing values (geom\_point).

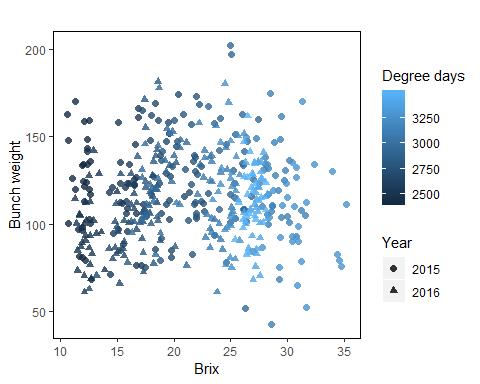


glimpse(mat\_tidy4)

## Observations: 450  
## Variables: 20  
## $ ID <chr> "2015\_101", "2015\_102", "2015\_103", "201...  
## $ sampling\_event <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...  
## $ Day <int> 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, ...  
## $ Month <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...  
## $ Year <int> 2015, 2015, 2015, 2015, 2015, 2015, 2015...  
## $ GFV <int> 2414, 2414, 2414, 2414, 2414, 2414, 2414...  
## $ Sample <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1...  
## $ POINT\_X <dbl> 286331.7, 286382.4, 286425.7, 286473.1, ...  
## $ POINT\_Y <dbl> 6236797, 6236810, 6236821, 6236833, 6236...  
## $ pH <dbl> 2.724, 2.685, 2.789, 2.801, 2.695, 2.744...  
## $ TA <dbl> 16.579, 16.699, 15.729, 15.304, 22.389, ...  
## $ brix <dbl> 12.1, 12.1, 12.4, 12.4, 10.7, 11.9, 12.0...  
## $ bunch\_wt <dbl> 121.55, 158.68, 148.42, 143.55, 100.64, ...  
## $ Mean\_berry\_wt <dbl> 1.153, 1.139, 1.268, 1.392, 1.023, 0.976...  
## $ berries\_bunch <int> 105, 139, 117, 103, 98, 99, 107, 106, 13...  
## $ Colour\_per\_berry\_Wt <dbl> 0.187, 0.204, 0.249, 0.218, 0.109, 0.134...  
## $ Phenolic\_per\_berry\_Wt <dbl> 1.156, 1.221, 1.142, 1.111, 1.126, 1.153...  
## $ Colour\_per\_berry <dbl> 0.214, 0.243, 0.332, 0.293, 0.116, 0.131...  
## $ Phenolic\_per\_berry <dbl> 1.324, 1.456, 1.523, 1.495, 1.194, 1.131...  
## $ Year\_a <fct> 2015, 2015, 2015, 2015, 2015, 2015, 2015...

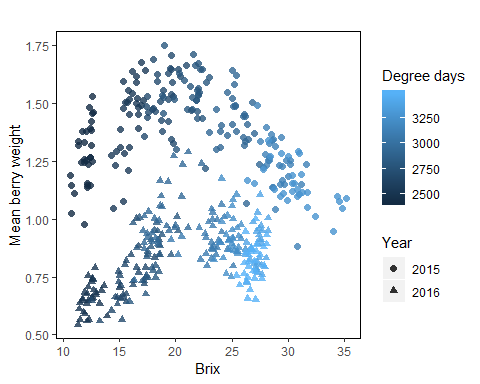
#Brix vs bunch\_wt  
mat\_tidy4 %>%  
 ggplot(aes(x = brix, y = bunch\_wt, colour = GFV, shape = Year\_a)) +  
 geom\_point(size = 2.0, alpha =0.8) +  
 labs(   
 x = "Brix",  
 y = "Bunch weight",  
 title = "",  
 colour = "Degree days",   
 shape = "Year") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_text(),  
 axis.text.y=element\_text(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

## Warning: Removed 1 rows containing missing values (geom\_point).



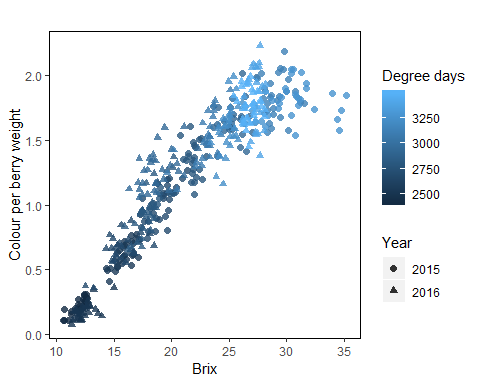
#Brix vs Mean\_berry\_wt  
mat\_tidy4 %>%  
 ggplot(aes(x = brix, y = Mean\_berry\_wt, colour = GFV, shape = Year\_a)) +  
 geom\_point(size = 2.0, alpha =0.8) +  
 labs(   
 x = "Brix",  
 y = "Mean berry weight",  
 title = "",  
 colour = "Degree days",   
 shape = "Year") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_text(),  
 axis.text.y=element\_text(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

## Warning: Removed 1 rows containing missing values (geom\_point).



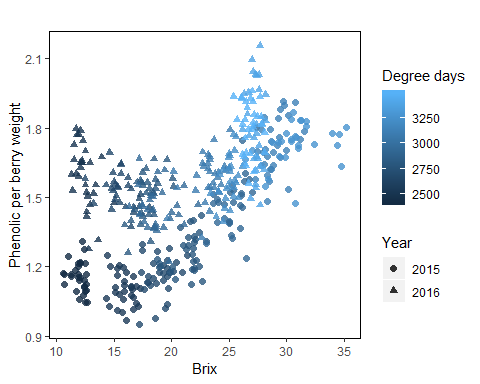
#Brix vs Colour\_per\_berry\_Wt  
mat\_tidy4 %>%  
 ggplot(aes(x = brix, y = Colour\_per\_berry\_Wt, colour = GFV, shape = Year\_a)) +  
 geom\_point(size = 2.0, alpha =0.8) +  
 labs(   
 x = "Brix",  
 y = "Colour per berry weight",  
 title = "",  
 colour = "Degree days",   
 shape = "Year") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_text(),  
 axis.text.y=element\_text(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

## Warning: Removed 1 rows containing missing values (geom\_point).



#Phenolic\_per\_berry\_Wt  
mat\_tidy4 %>%  
 ggplot(aes(x = brix, y = Phenolic\_per\_berry\_Wt, colour = GFV, shape = Year\_a)) +  
 geom\_point(size = 2.0, alpha =0.8) +  
 labs(   
 x = "Brix",  
 y = "Phenolic per berry weight",  
 title = "",  
 colour = "Degree days",   
 shape = "Year") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_text(),  
 axis.text.y=element\_text(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

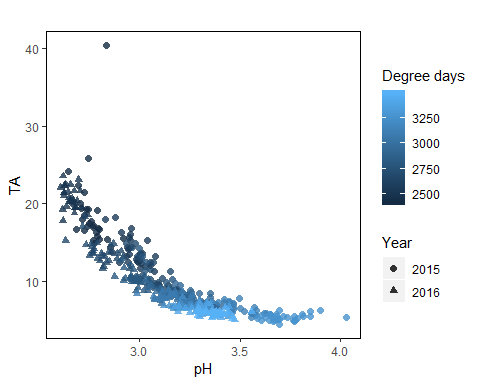
## Warning: Removed 1 rows containing missing values (geom\_point).



Do pH and TA have relationship?

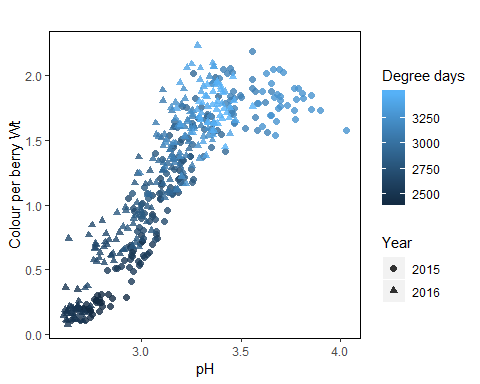
#pH vs TA  
mat\_tidy4 %>%  
 ggplot(aes(x = pH, y = TA, colour = GFV, shape = Year\_a)) +  
 geom\_point(size = 2.0, alpha =0.8) +  
 labs(   
 x = "pH",  
 y = "TA",  
 title = "",  
 colour = "Degree days",   
 shape = "Year") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_text(),  
 axis.text.y=element\_text(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

## Warning: Removed 1 rows containing missing values (geom\_point).



#pH vs colour  
mat\_tidy4 %>%  
 ggplot(aes(x = pH, y = Colour\_per\_berry\_Wt, colour = GFV, shape = Year\_a)) +  
 geom\_point(size = 2.0, alpha =0.8) +  
 labs(   
 x = "pH",  
 y = "Colour per berry Wt",  
 title = "",  
 colour = "Degree days",   
 shape = "Year") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_text(),  
 axis.text.y=element\_text(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

## Warning: Removed 1 rows containing missing values (geom\_point).



#phenolics vs colour  
mat\_tidy4 %>%  
 ggplot(aes(x = Phenolic\_per\_berry\_Wt, y = Colour\_per\_berry\_Wt, colour = GFV, shape = Year\_a)) +  
 geom\_point(size = 2.0, alpha =0.8) +  
 labs(   
 x = "Phenolic per berry Wt",  
 y = "Colour per berry Wt",  
 title = "",  
 colour = "Degree days",   
 shape = "Year") +  
 theme(panel.grid = element\_blank(), #remove panel lines  
 axis.text.x=element\_text(),  
 axis.text.y=element\_text(),  
 panel.background = element\_rect(fill = "white",  
 colour = "black"))

