

Grapevine

Naitong Chen, Shirley Cui, Shannon Edie

```
## Warning: package 'tidyverse' was built under R version 4.0.3
## -- Attaching packages ----- tidyverse 1
## v ggplot2 3.3.2    v purrr  0.3.4
## v tibble  3.0.3    v dplyr  1.0.4
## v tidyr   1.1.2    v stringr 1.4.0
## v readr   1.3.1    v forcats 0.5.0
## Warning: package 'tidyr' was built under R version 4.0.3
## Warning: package 'dplyr' was built under R version 4.0.3
## -- Conflicts ----- tidyverse_conflic
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## Warning: package 'corrplot' was built under R version 4.0.3
## corrplot 0.84 loaded
## Warning: package 'icenReg' was built under R version 4.0.3
## Loading required package: survival
## Loading required package: Rcpp
## Loading required package: coda
## Warning: package 'coda' was built under R version 4.0.3
## Warning: package 'lawstat' was built under R version 4.0.3
```

Problem overview

Agro-thermal heat-treatment technology in grapevines has been shown to increase yield, decrease the use of agro-chemicals used to fight pests, enhance wine quality and improve profits, but this technology has not been tested in the Okanagan Valley or Canada. Heat treatment is applied by driving a tractor through the vineyard rows that blows extreme heat into the canopy. During the growing seasons of 2019 and 2020, heat treatment was applied to a Merlot and a Chardonnay vineyard in the Okanagan valley six times during the growing season with application ten days apart from each other. Treatments were: (1) heat, and (2) control (no heat).

Data description

A full description of the recorded variables in the Chardonnay and Merlot studies can be seen in Figure 1. The blocking structure, sample sizes, and a brief overview of missing data are explained below.

Table 1: Variable descriptions.

Variable	Units	Definition and collection method	Grape	Year	Notes
# of clusters	Count	All clusters of each treatment vine were harvested in October of each year, and the clusters were counted for each vine.	Both	Both	
Yield	kg	All clusters of each treatment vine were harvested in October of each year and the clusters were weighed for each vine.	Both	Both	Possibly heteroscedastic for block in merlot
Average cluster weight per vine	kg/vine	Total grapevine yield divided by number of grape clusters.	Both	Both	Heteroscedastic for block in merlot

Heteroscedastic for treatment in chardonnay Possibly heteroscedastic for block in chardonnay | | Leaf greenness [SPAD] | Ten leaves per treatment vine (= subsample) were measured using a SPAD device in July, the number given in the data file is the average measurement. | Merlot | Both | | | 50% veraison | Days | 50% veraison is determined by estimating the % of berries on a treatment vine that have changed from green to red. This assessment is done three times and the 50% veraison is calculated via regression analysis. The raw data for 2019 is missing because berries were already red (100%) on the first assessment day and we could not determine the 50% value. The value represents the day of the month in August; numbers higher than 31 represent September. | Merlot | Both | Left-censored Possibly heteroscedastic for block in merlot | | Average berry weight per vine | kg / vine | Thirty berries were sampled in October from all clusters/vine (about 1-2 berries per cluster) and weighed. Average berry weight was calculated. | Merlot | Both | | | Average # of berries per cluster | Estimated count | Cluster weight (in grams) divided by the berry weight (in grams) | Merlot | Both | Heteroscedastic for block | | Berry quality: pH, TA, and Brix | Various | The same berries that were selected for weight were used by an analytical laboratory to determine berry TA (acidity), berry pH, and berry Brix, which can all influence the quality of the wine. | Merlot | Both | | | Brown seed color | Qualitative | Seed color changes according to grape maturity; assessed visually. This data is only available for 2020. | Merlot | 2020 | | | Pruning weight | kg | After dormancy and when the vines have lost all leaves, the canes were pruned in winter and weighed. | Merlot | 2019 | Heteroscedastic for treatment and block | | Ravaz index | Unitless | Yield divided by pruning weight. | Merlot | 2019 | | | Fruitfulness | Unitless | The number of clusters that emerged in Spring divided by the number of new shoots. | Merlot | 2020 | | | Bloom | Days | 50% bloom was determined by estimating the % of flowers on a treatment vine that have started to bloom. This assessment is done three times and the 50% bloom value calculated via regression analysis. The values represent June days (a value of 28 stands for June 28th, but values higher than 30 (=June 30st) are in July. | Merlot | 2020 | Left-censored Heteroscedastic for block |

There are 40 samples in each of the Chardonnay 2019 and Chardonnay 2020 datasets. The 40 samples consist of 8 samples from each of the 5 blocks. Among the 8 samples within each block, 4 samples of each treatment are included.

Both Chardonnay datasets contain 3 measurements of each sample with no missing data.

There are 60 samples in each of the Merlot 2019 and Merlot 2020 datasets. The 60 samples consist of 10 samples from each of the 6 blocks. Among the 10 samples within each block, 5 samples of each treatment are included.

In the Merlot 2019 data set, there are missing values from the measurement of veraison. In the Merlot 2020 data set, there are missing values from the measurements of veraison, bloom, yield, cluster weight, berries/cluster, berry weight, berry Brix, berry TA, berry pH, and brown seed color.

Handling missing data

As mentioned above, missing data was unique to the Merlot datasets.

Veraison

We know from the client that the veraison column indicates the number days into August in each respective year when 50% of the berries on a given vine had changed to red. The missing entries are a result of the vines reaching 50% veraison on the first day of measurement. They are missing not at random (MNAR) as whether this value is missing for each vine is dependent on this particular value. More specifically, if any of the missing values were recorded, it must be negative.

While imputation seems like a reasonable approach, since we know these values must be negative, we can take the maximum between each imputed value and zero to produce complete data sets?

Percentage of missing for the veraison measure of merlot2019 and merlot2020 are shown in Table 2.

Table 2: Percentage of Veraison data that is missing in the 2019 and 2020 Merlot datasets.

Missing.2019	Missing.2020	X...
43.3%	13.3	%

Bloom

Similar to veraison, the bloom column includes values that indicate the number of days into June in each respective year when 50% of the flowers on a given vine had started to bloom. The missing entries are a result of the vines reaching 50% bloom on the first day of measurement. They are also missing not at random (MNAR).

Again, we can consider taking the maximum between each imputed value and zero to produce complete data sets.

11.7% of the bloom data in the Merlot 2020 dataset is missing.

Yield related measures

For all columns that contain missing values (yield, cluster weight, berries/cluster, berry weight), the reason for missing is that the vine had no yield. Even though these missing values depend on the measures of cluster number of their corresponding vines being zero, these yield related values are missing precisely because there are no yield. This is then also a case of missing not at random (MNAR).

Assigning these missing entries the value of zero seems plausible.

Percentage of missing for the yield related measures of merlot2020 is shown below.

```
## [1] 0.1166667
```

Berry related measures

Similarly, for all columns that contain missing values (berry Brix, berry TA, berry pH, brown seed color), the reason for missing is that the vine had no yield. These values are missing simply because there were no grapes produced to have these measurements recorded. Depending on the relationship between berry quality and yield, this is either a case of missing at random (MAR, in the absence of an association between berry quality and yield) or missing completely at random (MCAR, in the presence of an association between berry quality and yield).

Due to the structure of the study, particularly the existence of blocking and replication, imputation seems to be the most reasonable approach to fill in the missing values.

Percentage of missing for the berry related measures of merlot2020 is shown below.

```
## [1] 0.1333333
```

Question

In the merlot2020 data set, most of the vines that had no entries for bloom or veraison (implying earlier ripening) also had no yield. This is against my intuition. If anything, as observed in the other three data sets, early ripening may be associated with higher yield? What was the weather event that caused some of the vines to have no yield?

Constructing complete data sets

We construct complete data sets using the approaches described above. Once these complete data sets are constructed, analysis can be done separately on each of the imputed data sets. Although we need to look into the necessity and methods to aggregate these results. We should also look into the order or columns to impute, as well as whether all responses should be used to predict the missing values.

For now, I have imputed with and without the yield related measures. We may consider more variants such as using only responses that are highly correlated.

Linear regressions are used for imputation. There are a lot more available options specified on page 76 of <https://cran.r-project.org/web/packages/mice/mice.pdf>.

Note that the the columns of key, row and subsample are excluded from all imputations below.

merlot2019

The imputed values of veraison for merlot2019 from the first imputed data set are shown below. Note they are all above zero, they will all be filled in with zero before the analysis.

```
## [1] 21.715994 18.125289 16.773634 14.942121 19.471914 21.589641 21.486780
## [8] 15.115459 8.349683 18.339226 20.047271 13.725877 21.528018 7.691104
## [15] 18.035914 20.088900 22.080232 11.933042 13.388718 2.451018 8.835080
## [22] 24.841542 25.660091 20.944922 9.503893 11.462480
```

merlot2020

The first set of imputed values for merlot2020 using all responses are shown below.

```
## [1] "veraison"
## [1] 30.67912 30.34631 31.56788 30.22332 25.51688 28.10610 27.17114 28.72719
## [1] "berry Brix"
## [1] 27.28276 27.34771 25.86445 27.62986 28.48672 28.04230 27.84948 28.06373
## [1] "berry pH"
## [1] 3.519315 3.523890 3.232155 3.496757 3.630889 3.655843 3.511203 3.484332
## [1] "berry TA"
## [1] 6.349962 6.422805 8.139608 6.498603 5.441846 4.693712 6.532854 6.638227
## [1] "brown seed color"
## [1] 63.62264 63.88353 75.09464 64.83935 72.78476 73.82176 68.46240 68.89437
```

The first set of imputed values for merlot2020 not using yield related measures are shown below.

```
## [1] "veraison"
## [1] 32.35104 31.94097 31.91933 31.39454 27.36092 29.21802 25.75810 30.42609
## [1] "berry Brix"
## [1] 26.22747 26.31896 25.91935 26.55039 27.50730 26.92197 26.82877 27.06777
## [1] "berry pH"
## [1] 3.290631 3.298345 3.231982 3.272862 3.408987 3.421944 3.281947 3.258690
## [1] "berry TA"
## [1] 7.357714 7.404429 8.037524 7.580077 6.348228 5.812225 7.558876 7.710729
## [1] "brown seed color"
## [1] 79.30215 79.52354 76.95949 78.53604 89.29272 88.31270 84.29497 83.44413
```

Again the veraison values are all above zero, and so zero values will assigned before conducting the analysis.

We therefore won't need to do imputation on veraison altogether, and so can proceed for now using the following three sets of data (merlot2019, merlot2020 imputed with filled-in zeros, merlot2020 imputed without filled-in zeros).

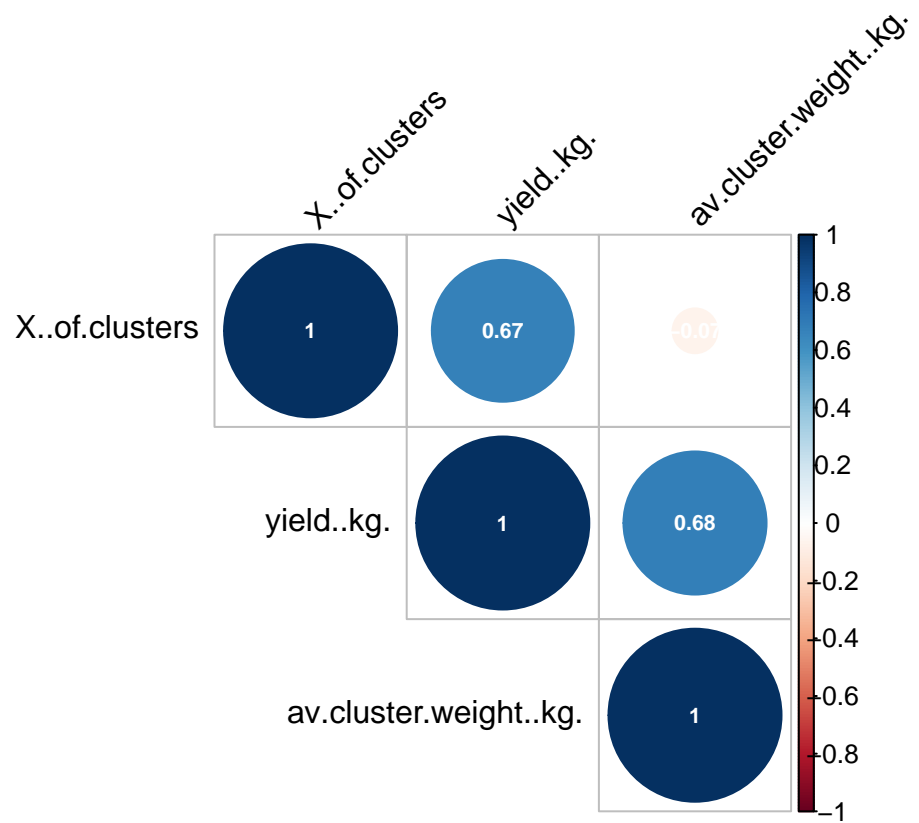
Check out the imputed data sets in the code blocks!

data preprocessing - PCA

Basically, we have reduced the dimensions of the dataset down to four/three dimensions for merlot 2013/2014 using PCA.

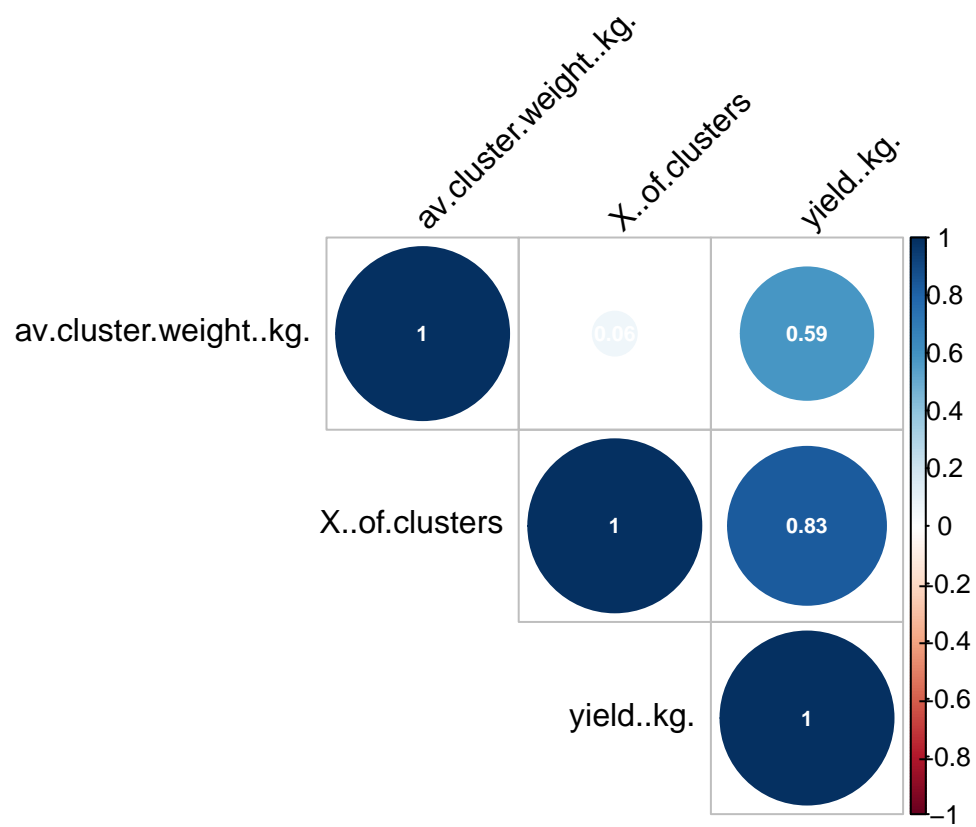
Check for correlations

chardonnay 2019



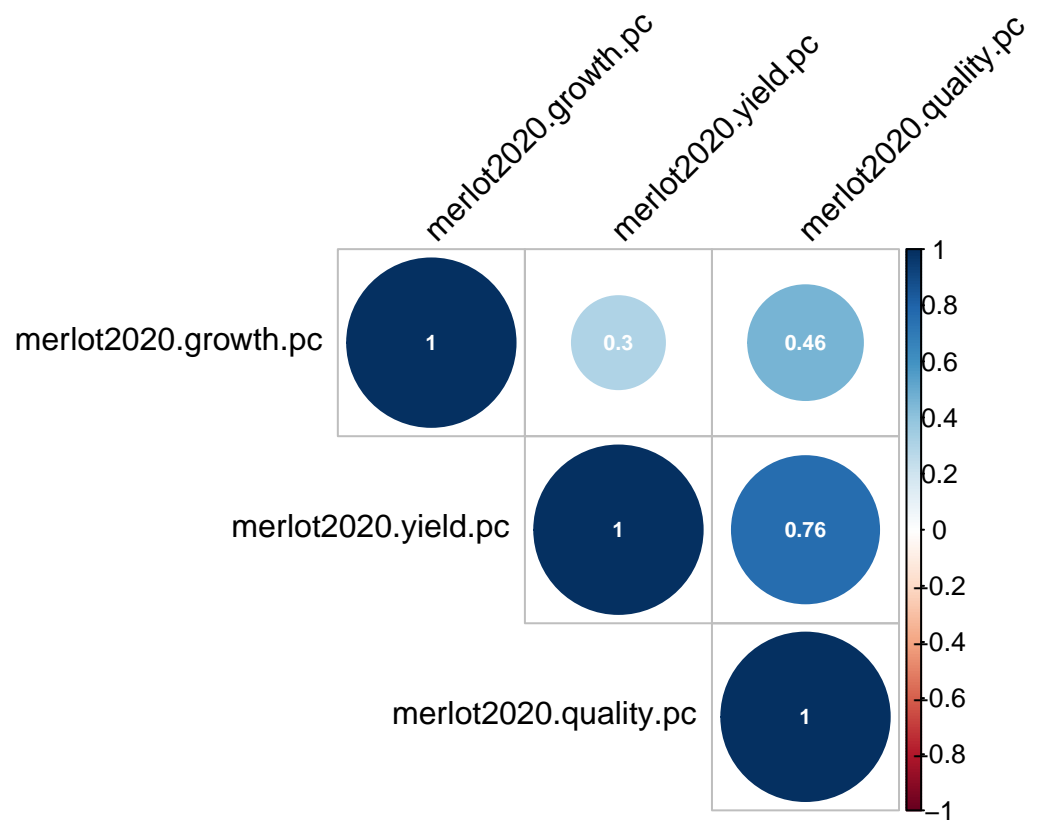
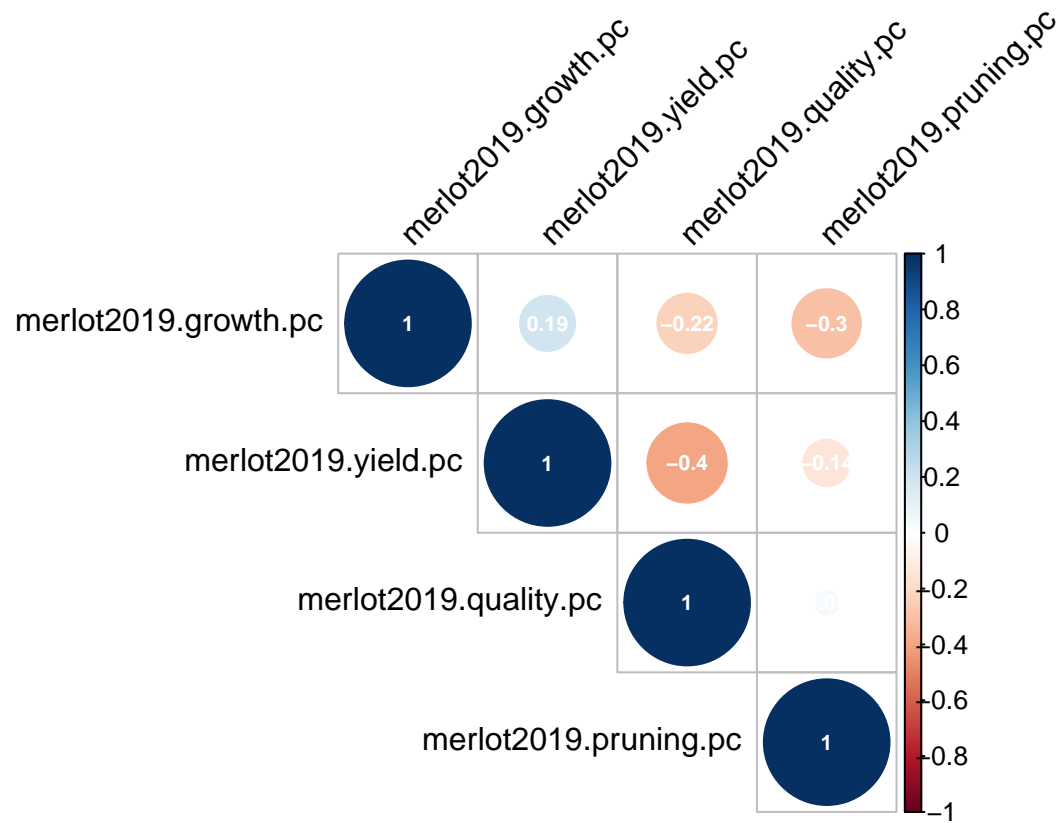
From the output of correlation function, there seem no correlations between response variables.

chardonnay 2020



Yield is found to be positively correlated with number of clusters.

merlot 2019



merlot 2020

From the correlation matrix plots, there seem to be no obvious correlations among the response variables in these four datasets. We are good to continue with normality and equal variance assumptions check.

Check for assumption of ANOVA test: normality and equal variance

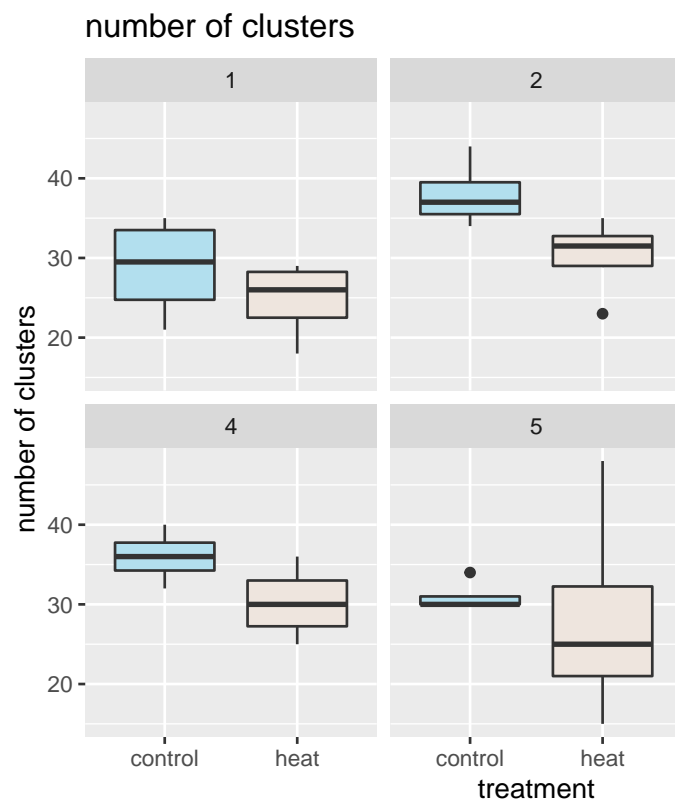
Chardonnay 2019

1. table of mean and variance under control and treatment for each response variable

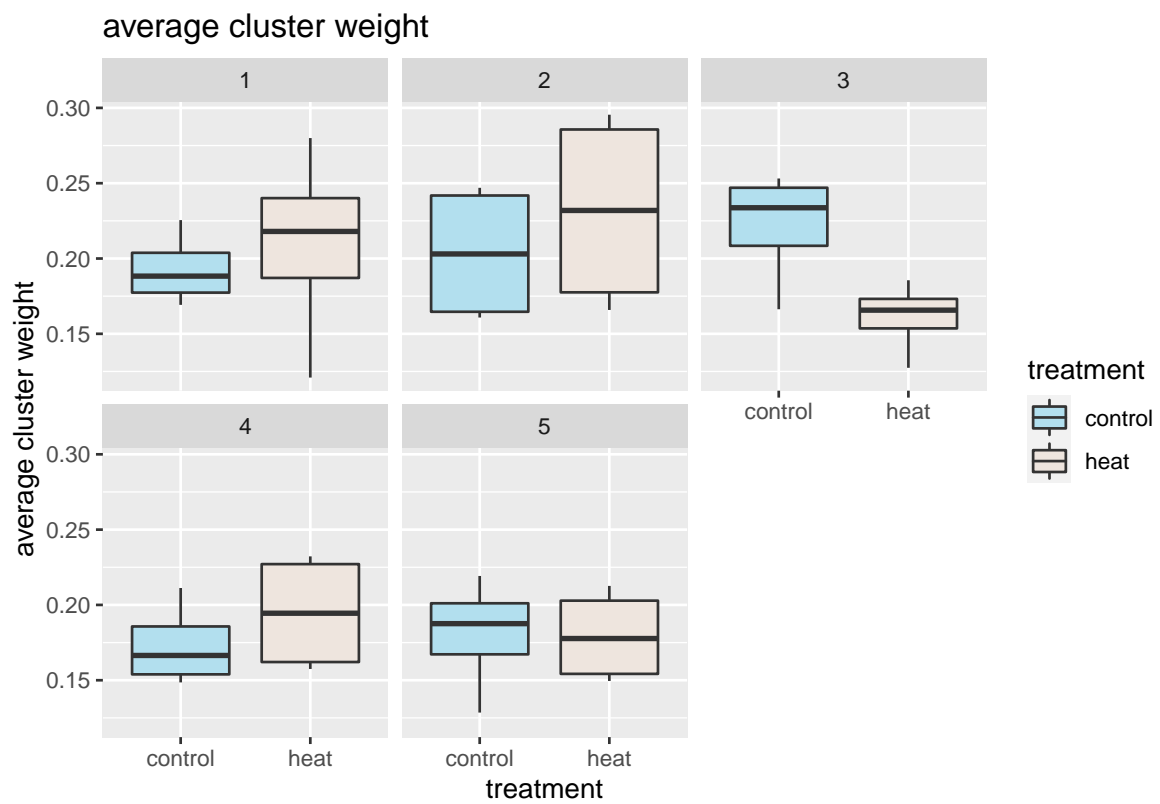
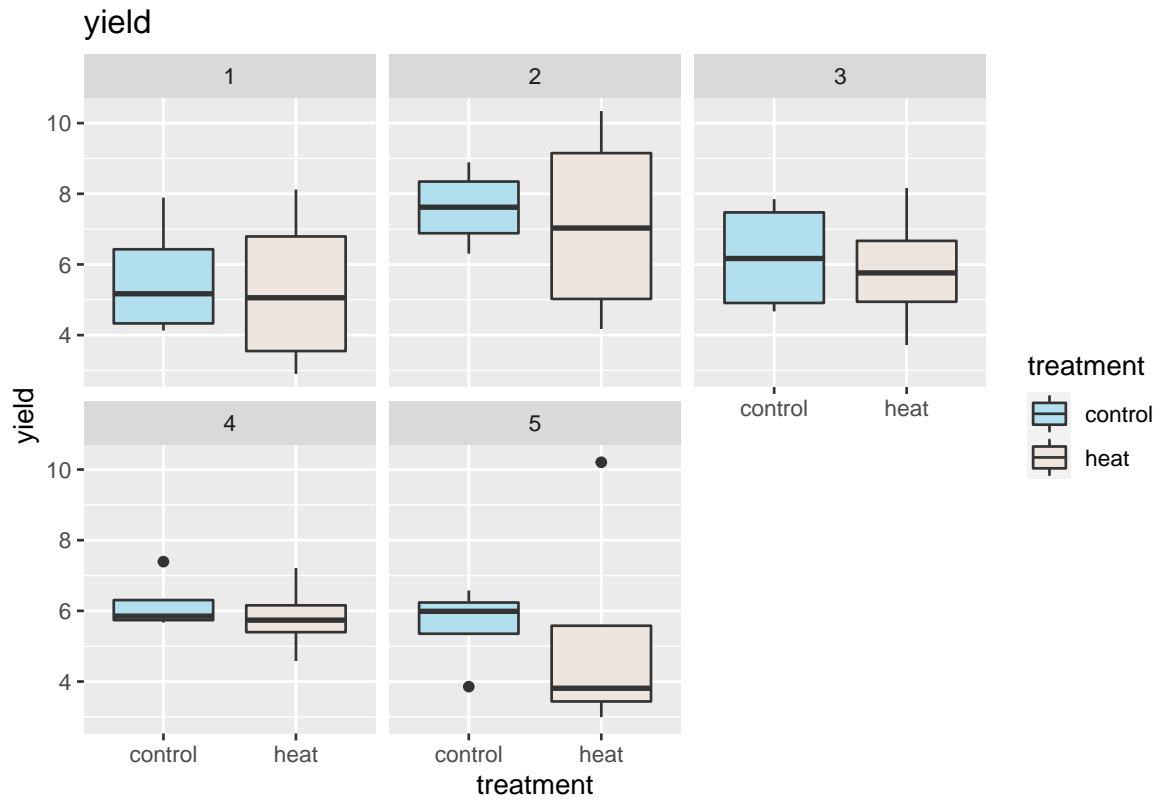
`summarise()` has grouped output by 'block'. You can override using the `.groups` argument.

Table 3: Summary statistics for outcome variables in the Chardonnay 2019 dataset.

block	treatment	var.num_of_clusters	mean.num_of_clusters	yield	mean.yield	var.cluster_weight	mean.cluster_weight
1	control	41.58333	28.75	2.9934334	5.590526	0.0006005	0.1928360
1	heat	24.91667	24.75	5.7176824	5.284351	0.0043685	0.2092217
2	control	18.66667	38.00	1.3118024	7.609012	0.0021549	0.2034579
2	heat	26.25000	30.25	8.3402586	7.144080	0.0045004	0.2312906
3	control	22.00000	28.00	2.6102314	6.214215	0.0015320	0.2217168
3	heat	99.66667	36.50	3.4153842	5.851342	0.0005988	0.1611007
4	control	11.33333	36.00	0.6549582	6.191536	0.0007924	0.1731682
4	heat	22.91667	30.25	1.1650369	5.817322	0.0015669	0.1946832
5	control	4.00000	31.00	1.4450230	5.601866	0.0014714	0.1807254
5	heat	198.25000	28.25	11.29734355	5.204972	0.0009870	0.1793797



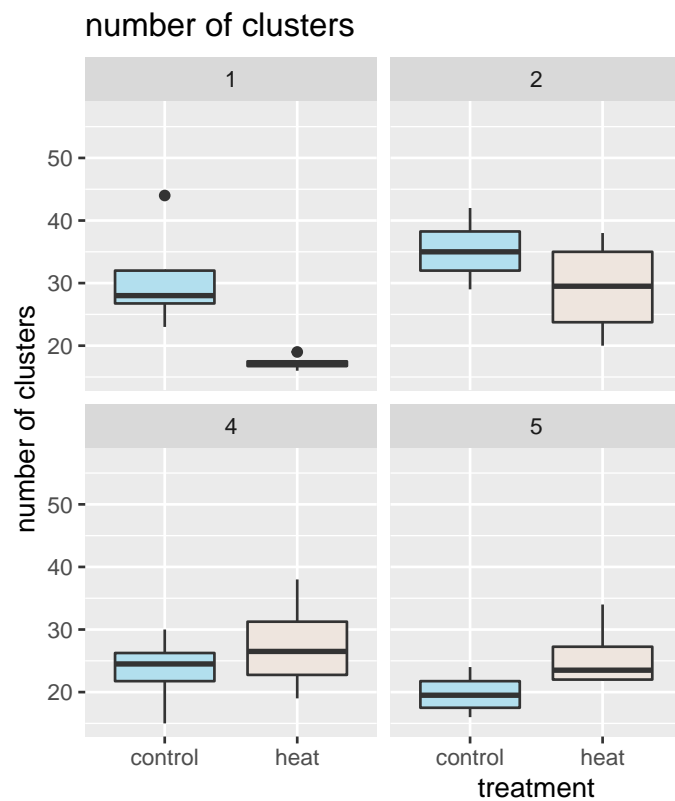
2. Visualization (side by side boxplots for each response variable)



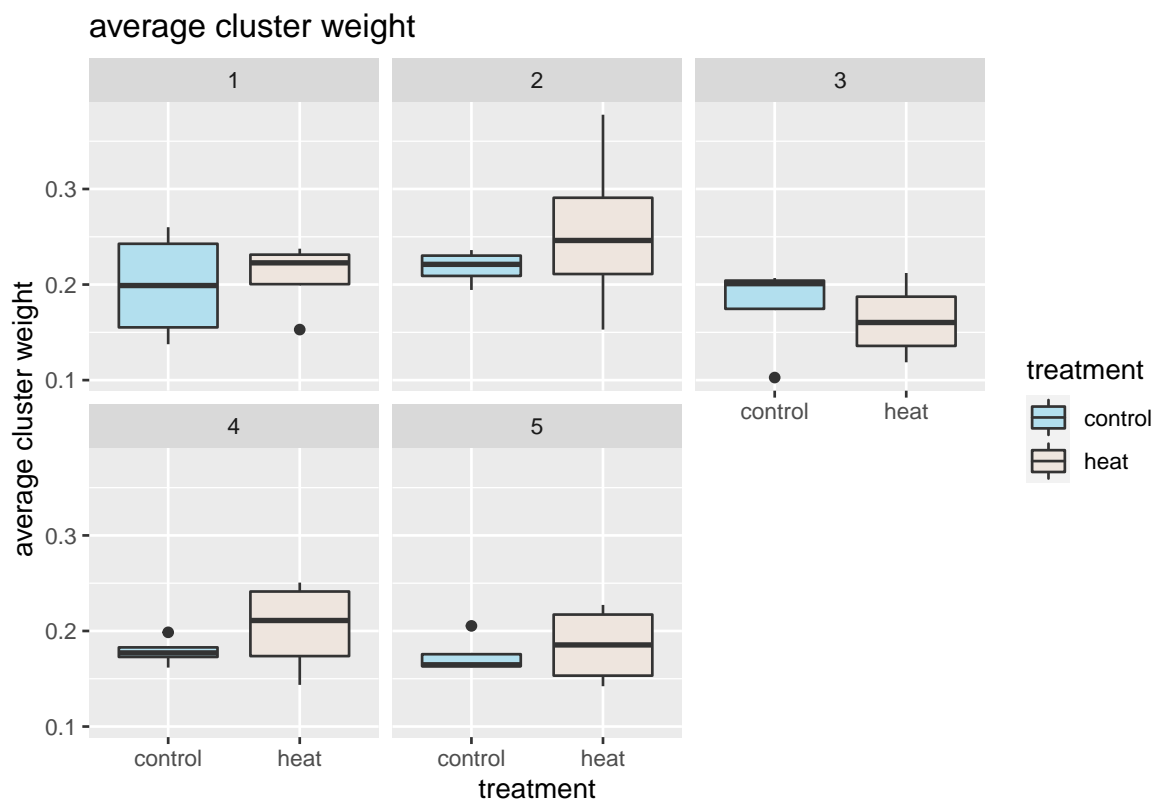
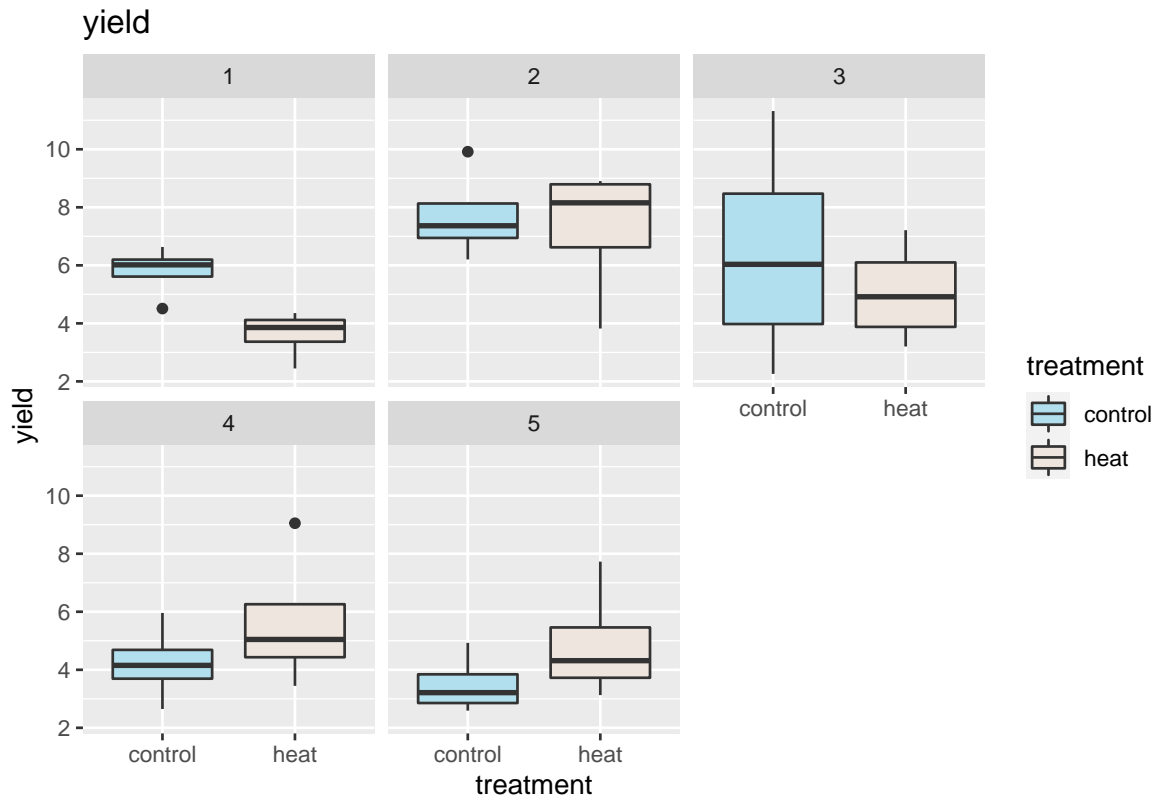
Chardonnay 2020

3. table of mean and variance under control and treatment for each response variable

```
## `summarise()` has grouped output by 'block'. You can override using the `.groups` argument.
## # A tibble: 10 x 8
## # Groups:   block [5]
##   block treatment var.num_of_clus~ mean.num_of_clu~ var.yield mean.yield
##   <fct> <fct>          <dbl>          <dbl>    <dbl>    <dbl>
## 1 1 control          83.6            30.8    0.818    5.79
## 2 1 heat             1.58            17.2    0.699    3.63
## 3 2 control          30.9            35.2    2.48     7.71
## 4 2 heat            67.6            29.2    5.61     7.26
## 5 3 control          275            34.5    15.3     6.41
## 6 3 heat             9.67            30.5    3.15     5.06
## 7 4 control          39            23.5    1.84     4.23
## 8 4 heat            65.7            27.5    5.77     5.65
## 9 5 control          12.2            19.8    1.06     3.49
## 10 5 heat            32.2            25.8    4.04     4.87
## # ... with 2 more variables: var.cluster_weight <dbl>,
## #   mean.cluster_weight <dbl>
```



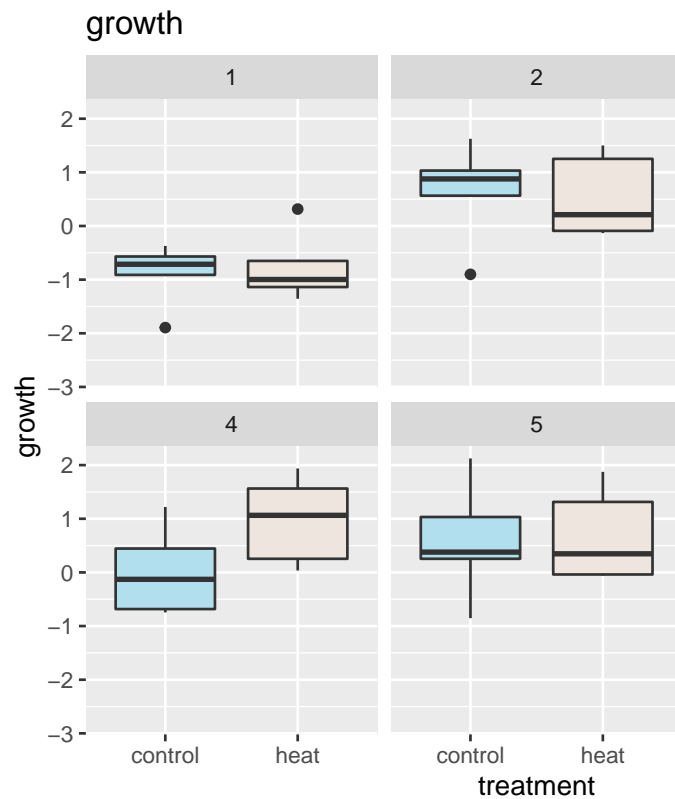
2. Visualization (side by side boxplots for each response variable)



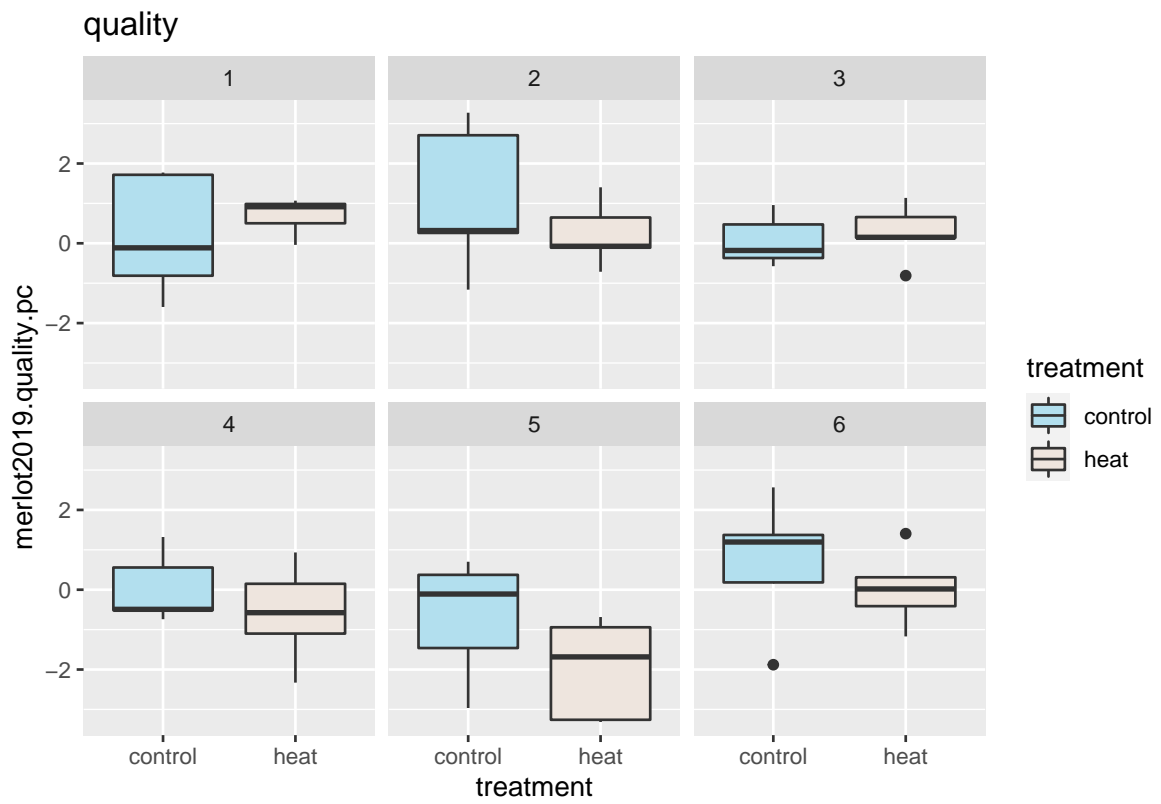
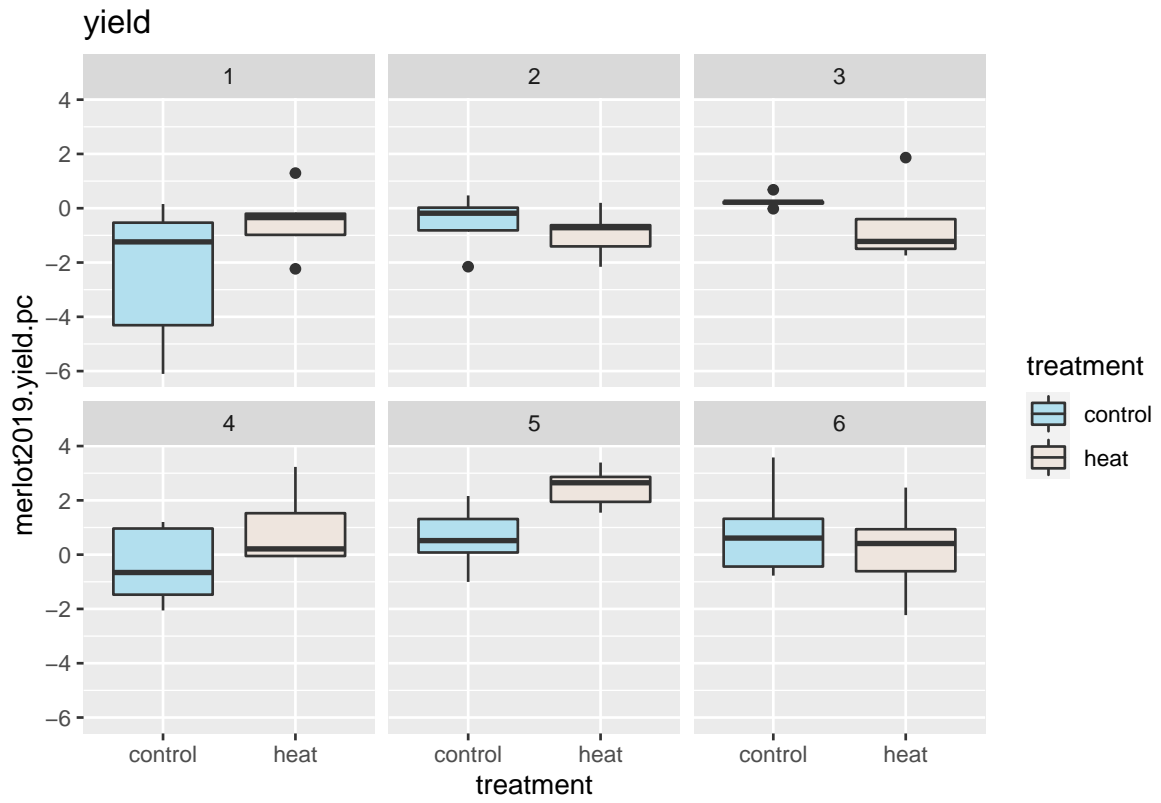
merlot 2019

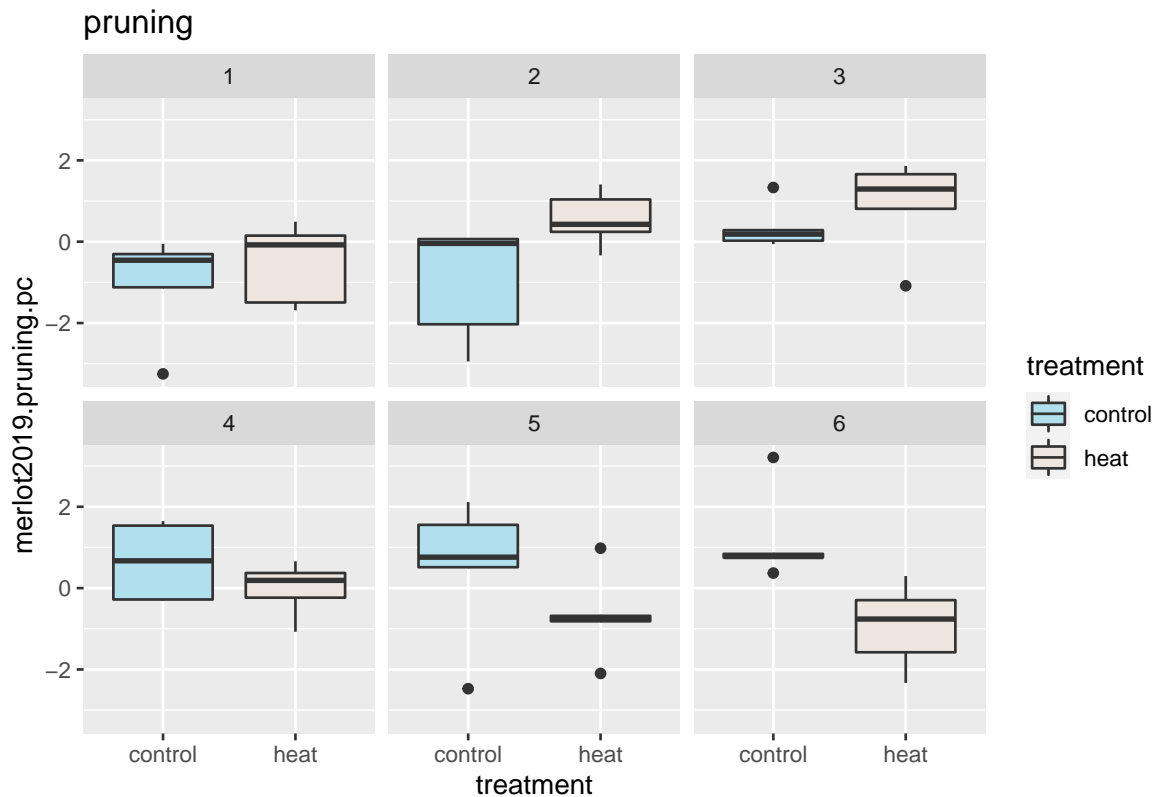
3. table of mean and variance under control and treatment for each response variable

```
## `summarise()` has grouped output by 'block'. You can override using the `.groups` argument.
## # A tibble: 12 x 8
## # Groups:   block [6]
##   block treatment var.growth mean.growth var.yield mean.yield var.quality
##   <dbl> <fct>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 1 control    0.353    -0.892    1.69    -1.04    2.28
## 2 1 heat      0.431    -0.765    0.995   -0.524    0.212
## 3 2 control    0.891     0.640    2.02   -0.974    3.43
## 4 2 heat      0.597     0.548    0.468    0.556    0.662
## 5 3 control    1.39    -0.139    0.318    0.353    0.405
## 6 3 heat      0.277    -0.103    1.40     0.909    0.525
## 7 4 control    0.681     0.0216   0.889    0.655    0.776
## 8 4 heat      0.671     0.971    0.453   -0.0178   1.54
## 9 5 control    1.20     0.587    3.16     0.494    2.29
##10 5 heat      0.751     0.688    1.20   -0.670    1.57
##11 6 control    0.767    -1.65     1.31     1.19    2.77
##12 6 heat      0.546     0.0970   1.08   -0.934    0.902
## # ... with 1 more variable: mean.quality <dbl>
```



2. Visualization (side by side boxplots for each response variable)



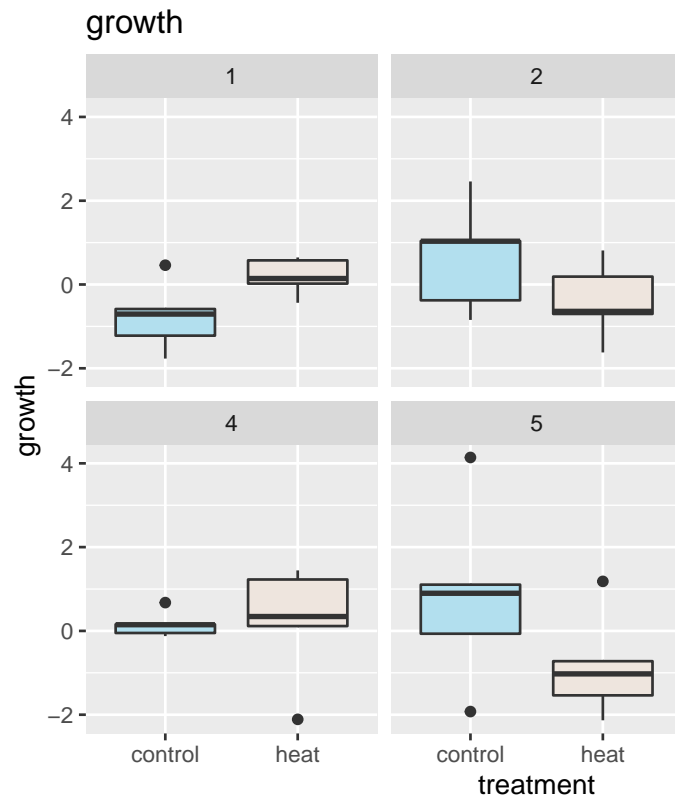


merlot 2020

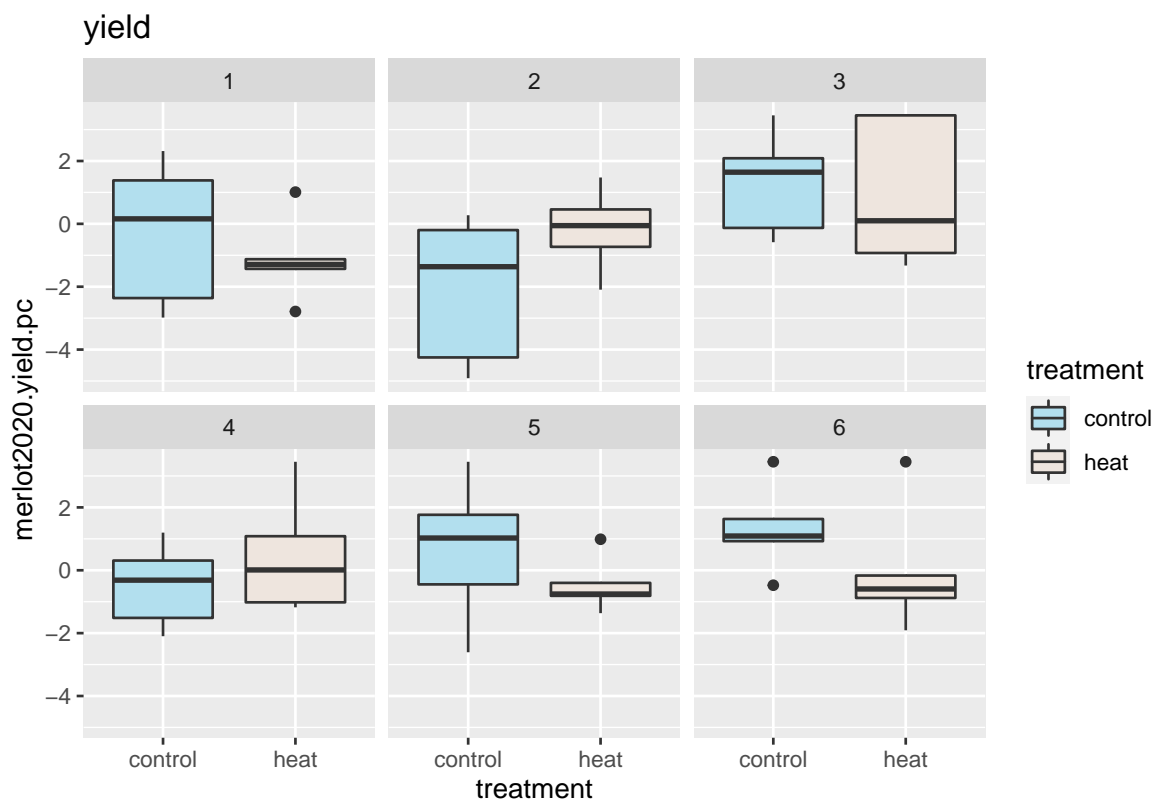
1. table of mean and variance under control and treatment for each response variable

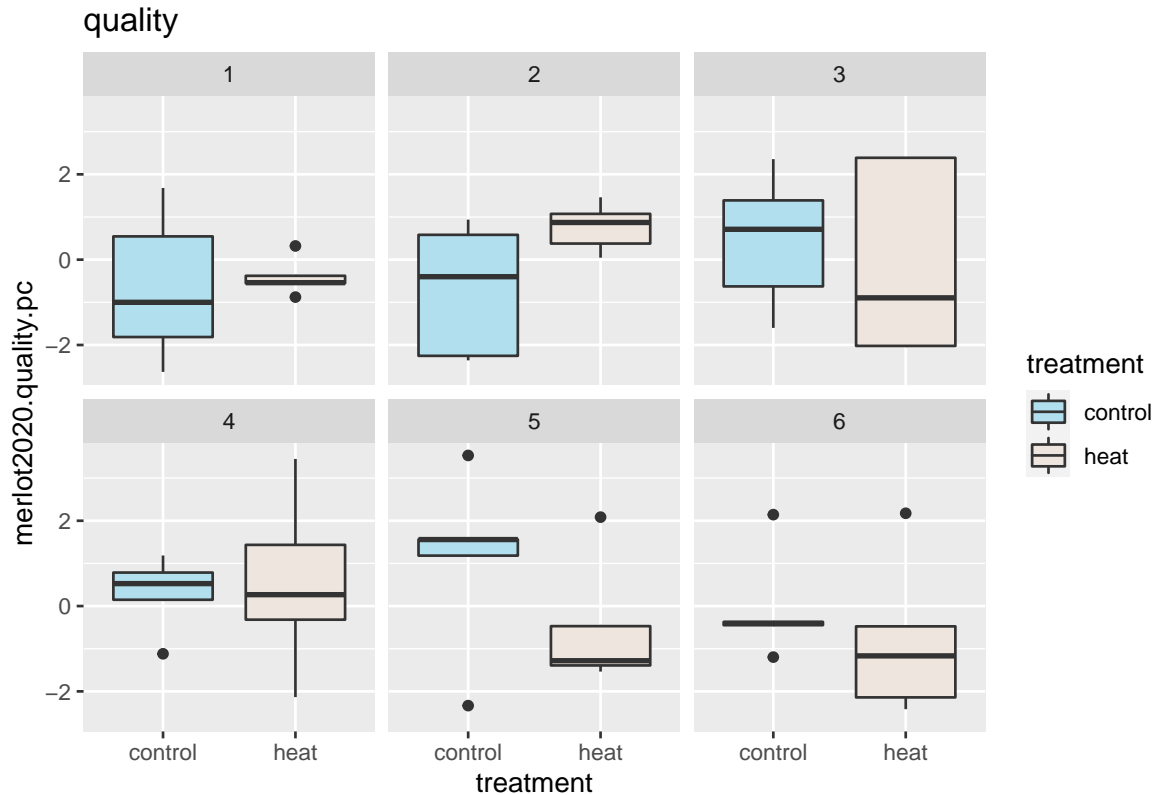
`summarise()` has grouped output by 'block'. You can override using the `.groups` argument.

```
## # A tibble: 12 x 8
## # Groups:   block [6]
##   block treatment var.growth mean.growth var.yield mean.yield var.quality
##   <fct> <fct>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 1 control 0.688 -0.764 5.33 -0.297 3.06
## 2 1 heat 0.195 0.190 1.86 -1.13 0.198
## 3 2 control 1.72 0.661 5.58 -2.09 2.40
## 4 2 heat 0.865 -0.393 1.78 -0.190 0.315
## 5 3 control 3.88 0.567 2.74 1.29 2.49
## 6 3 heat 0.0703 -0.445 5.49 0.950 5.11
## 7 4 control 0.0977 0.161 1.79 -0.484 0.777
## 8 4 heat 1.99 0.204 3.61 0.469 4.31
## 9 5 control 4.86 0.830 5.26 0.637 4.53
## 10 5 heat 1.57 -0.848 0.782 -0.470 2.29
## 11 6 control 0.565 0.641 2.02 1.32 1.63
## 12 6 heat 0.388 -0.805 4.18 -0.0205 3.38
## # ... with 1 more variable: mean.quality <dbl>
```



2. Visualization (side by side boxplots for each response variable)





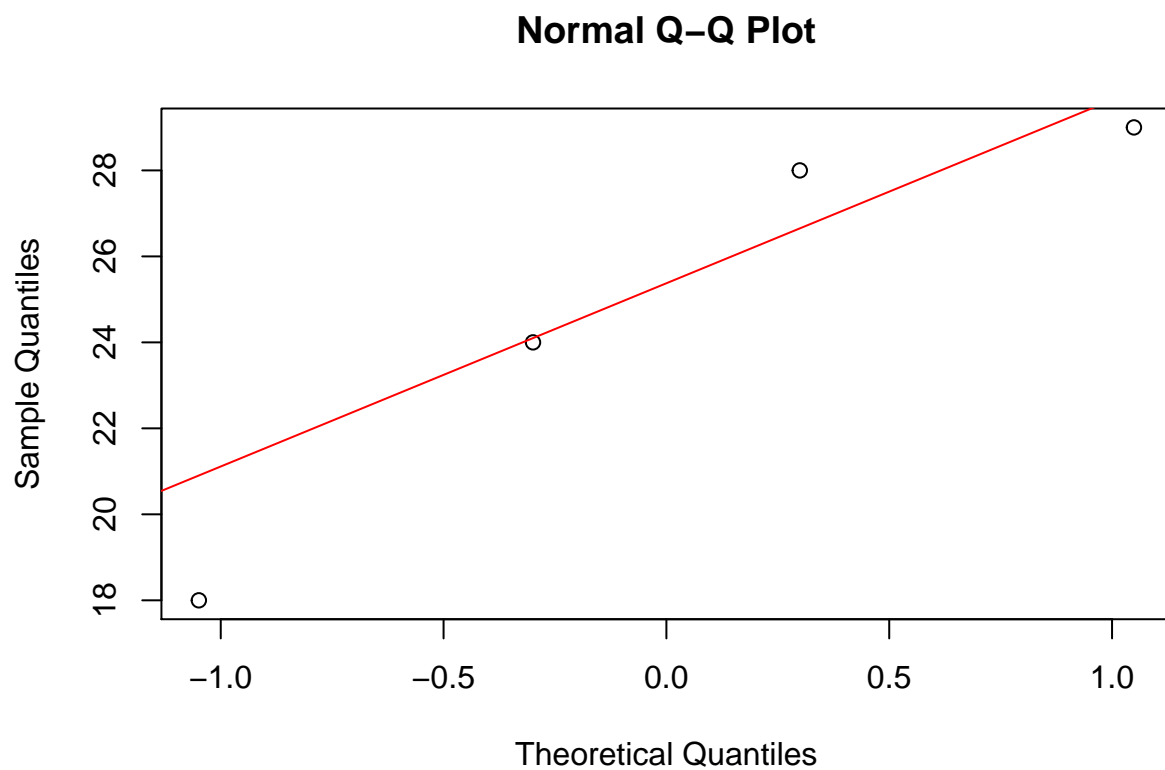
Based on the side-by-side box plots above, it seems that many outcome variables do not have equal variance, or the distribution is not quite normal. One limitation which can explain this is that our sample size is really small for each treatment in each block (5 or 6 data points). Hence, the plot cannot really tell much useful information about the data.

Below, wilcoxon rank tests can be performed to test whether the distribution in control and treatment groups are the same. (that would be a lot of tests... I only tried out the test for number of clusters in Char2019 for five blocks.) ### randomization test chardonnay 2019

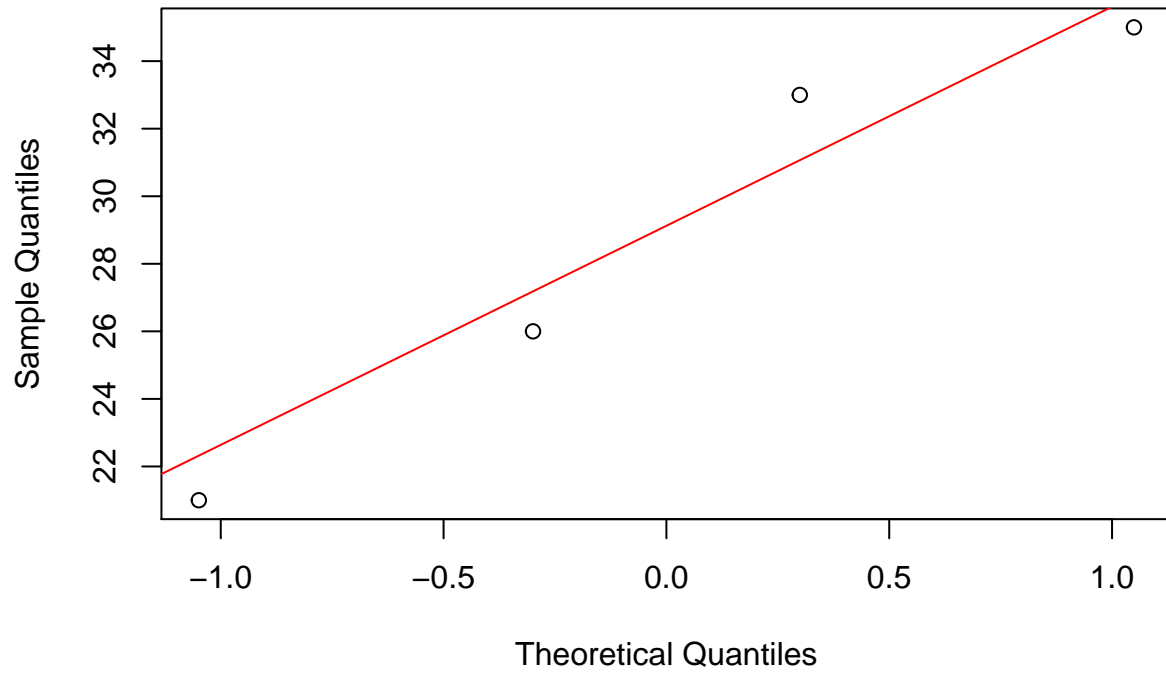
```
##
## Wilcoxon rank sum test with continuity correction
##
## data: chardonnay2019$X..of.clusters[chardonnay2019$block == i & chardonnay2019$treatment == "heat"]
## W = 5, p-value = 0.4705
## alternative hypothesis: true location shift is not equal to 0
##
##
## Wilcoxon rank sum test with continuity correction
##
## data: chardonnay2019$X..of.clusters[chardonnay2019$block == i & chardonnay2019$treatment == "heat"]
## W = 1, p-value = 0.0606
## alternative hypothesis: true location shift is not equal to 0
##
##
## Wilcoxon rank sum test with continuity correction
##
## data: chardonnay2019$X..of.clusters[chardonnay2019$block == i & chardonnay2019$treatment == "heat"]
## W = 13, p-value = 0.1913
## alternative hypothesis: true location shift is not equal to 0
```

```
##
##
## Wilcoxon rank sum test with continuity correction
##
## data:  chardonnay2019$X..of.clusters[chardonnay2019$block == i & chardonnay2019$treatment == "heat"]
## W = 2.5, p-value = 0.1465
## alternative hypothesis: true location shift is not equal to 0
##
##
## Wilcoxon rank sum test with continuity correction
##
## data:  chardonnay2019$X..of.clusters[chardonnay2019$block == i & chardonnay2019$treatment == "heat"]
## W = 4, p-value = 0.3005
## alternative hypothesis: true location shift is not equal to 0
```

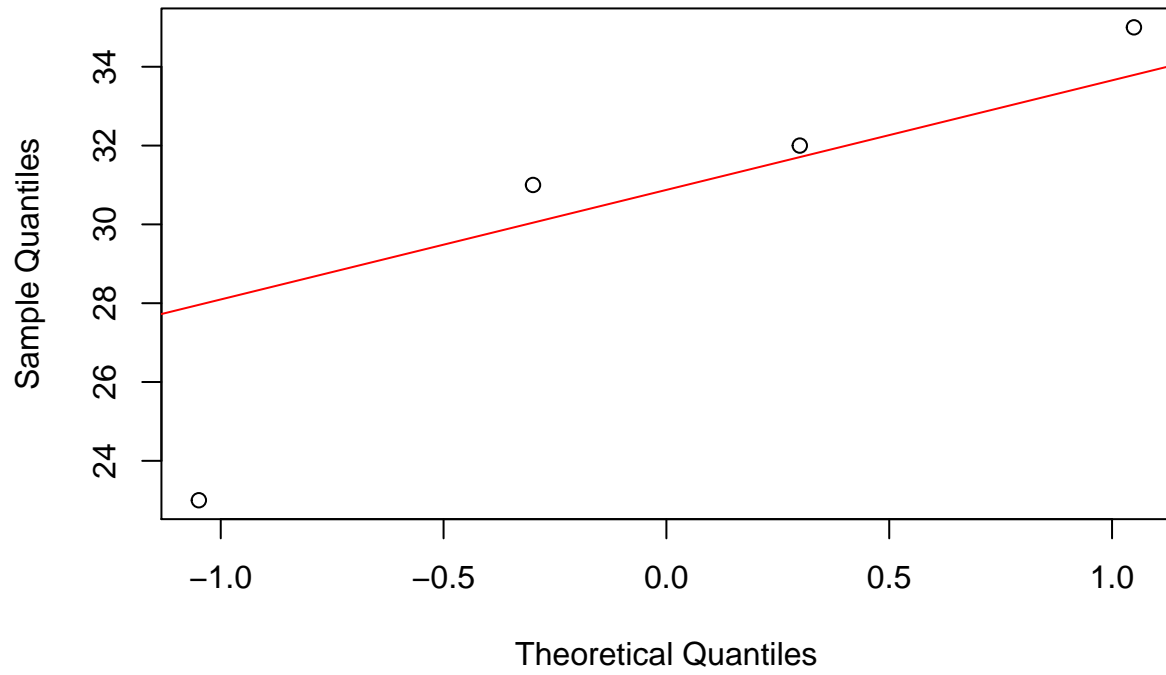
QQ plot for number of clusters in chardonnary 2019 (for each block/treatment combo)



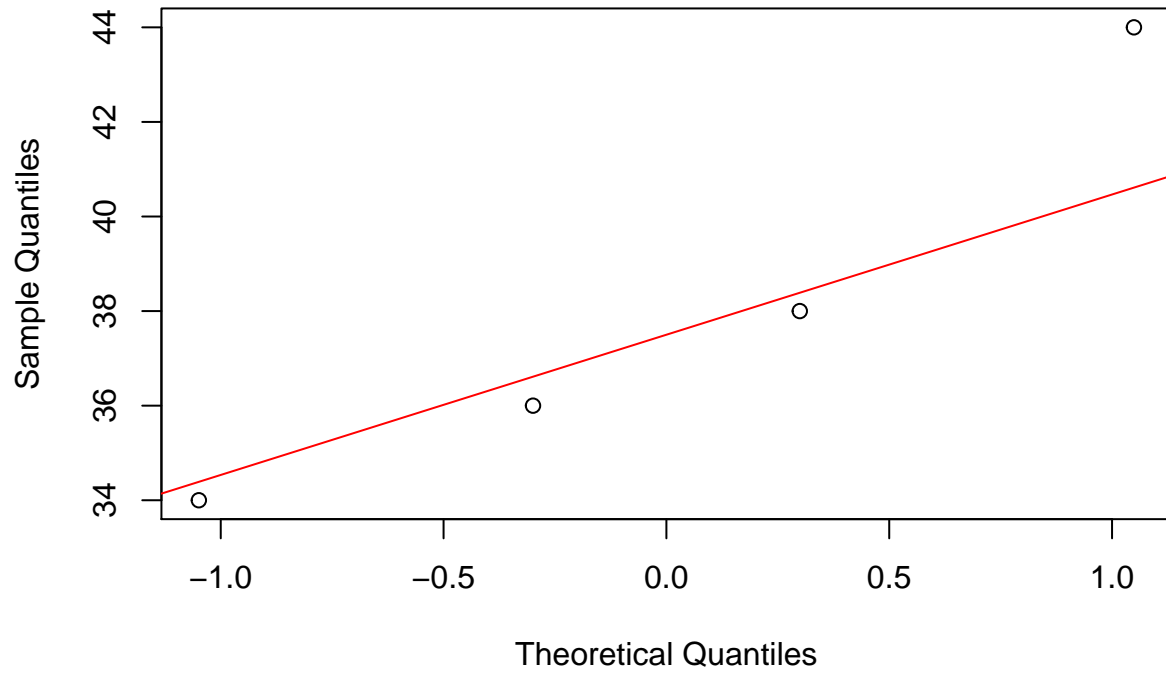
Normal Q-Q Plot



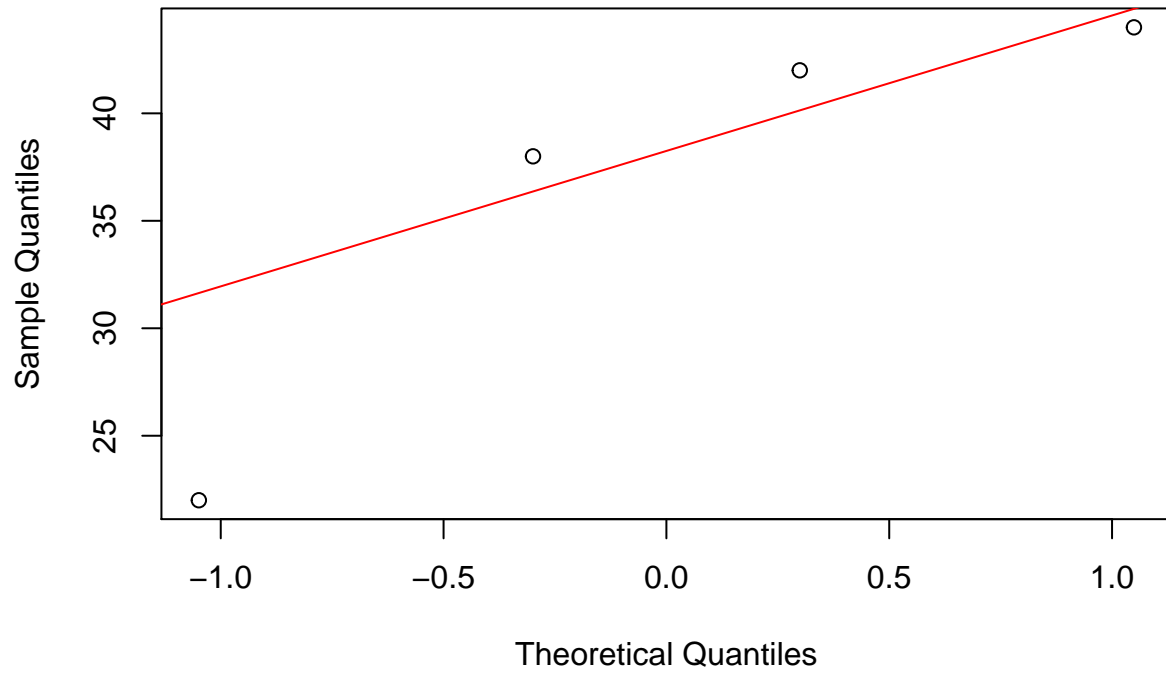
Normal Q-Q Plot

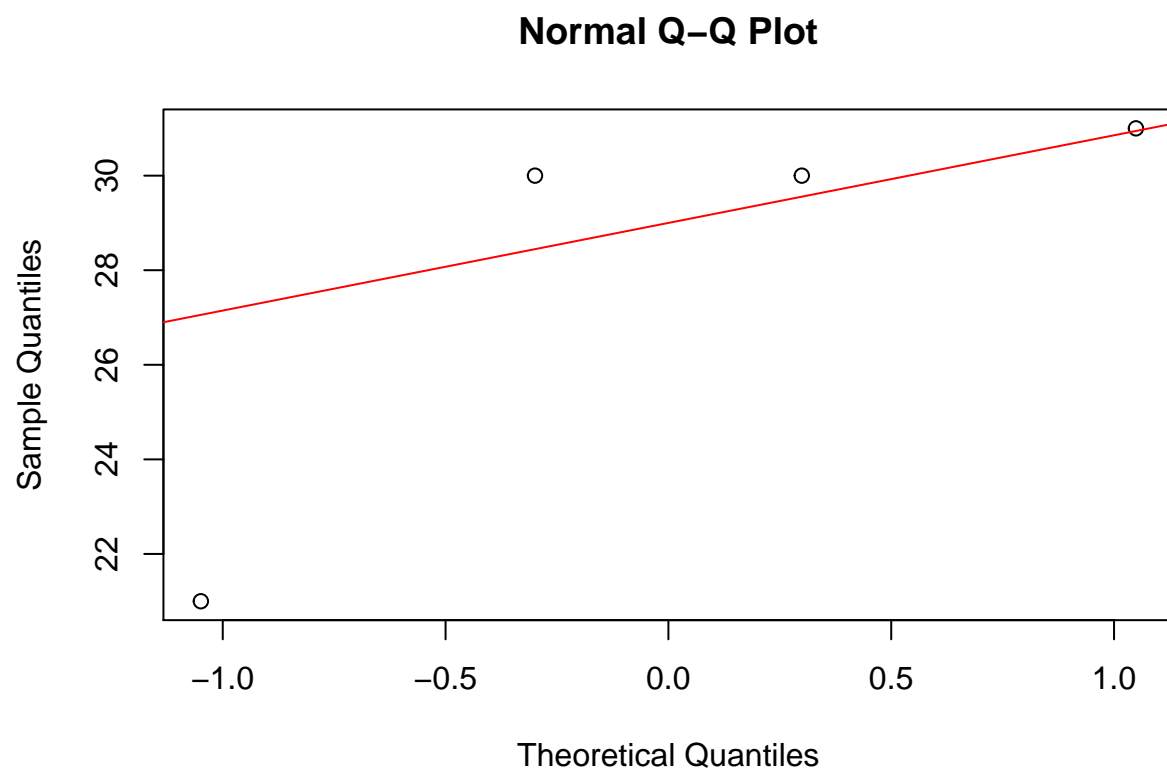


Normal Q-Q Plot

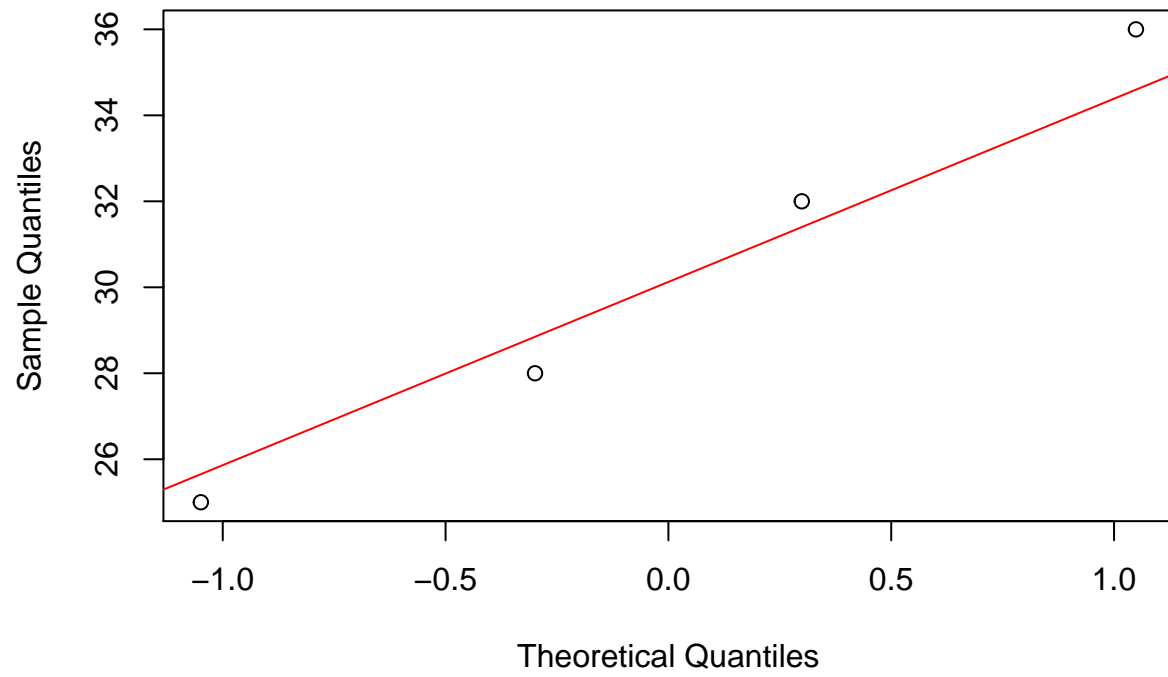


Normal Q-Q Plot

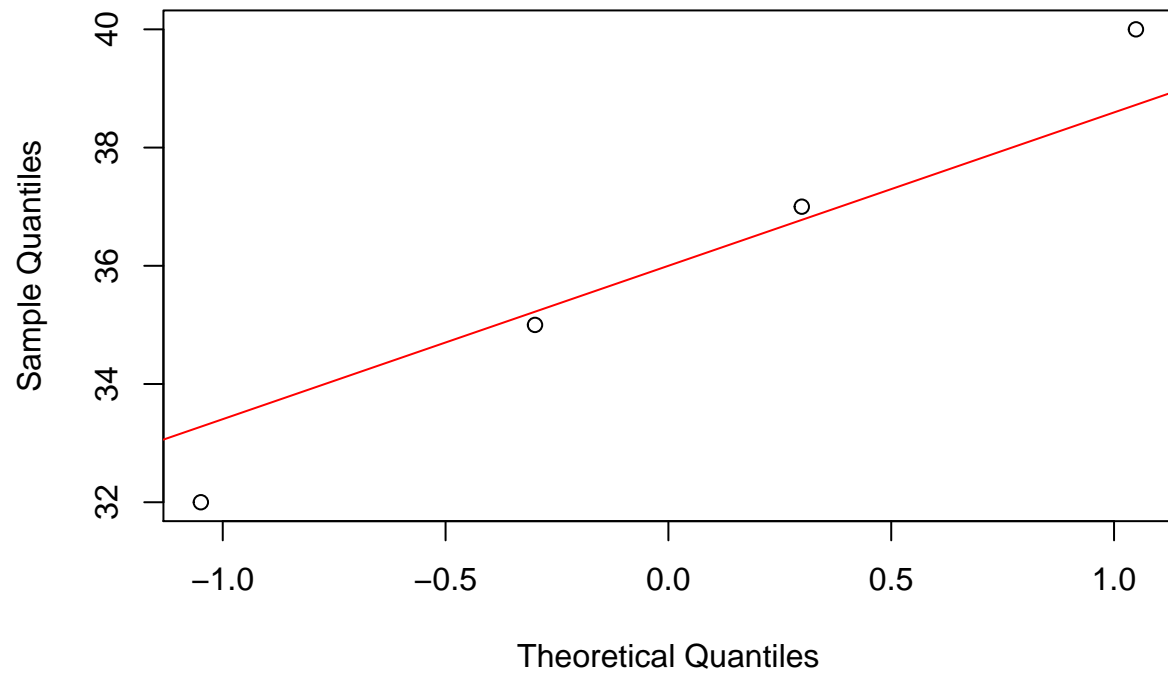




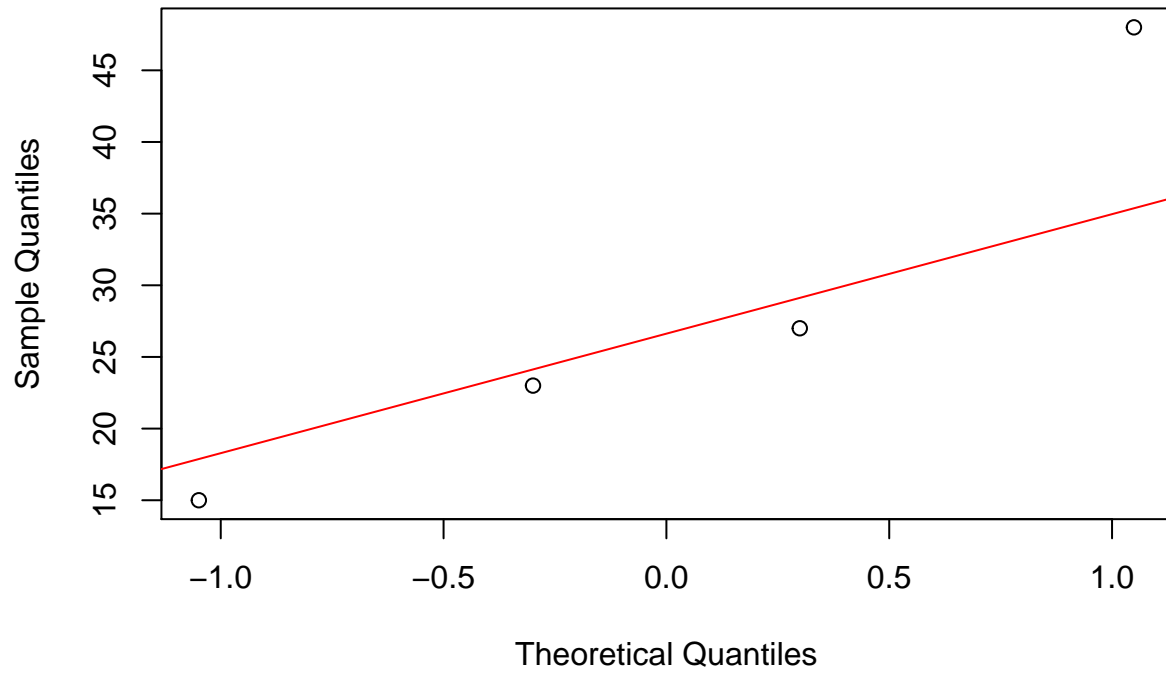
Normal Q-Q Plot

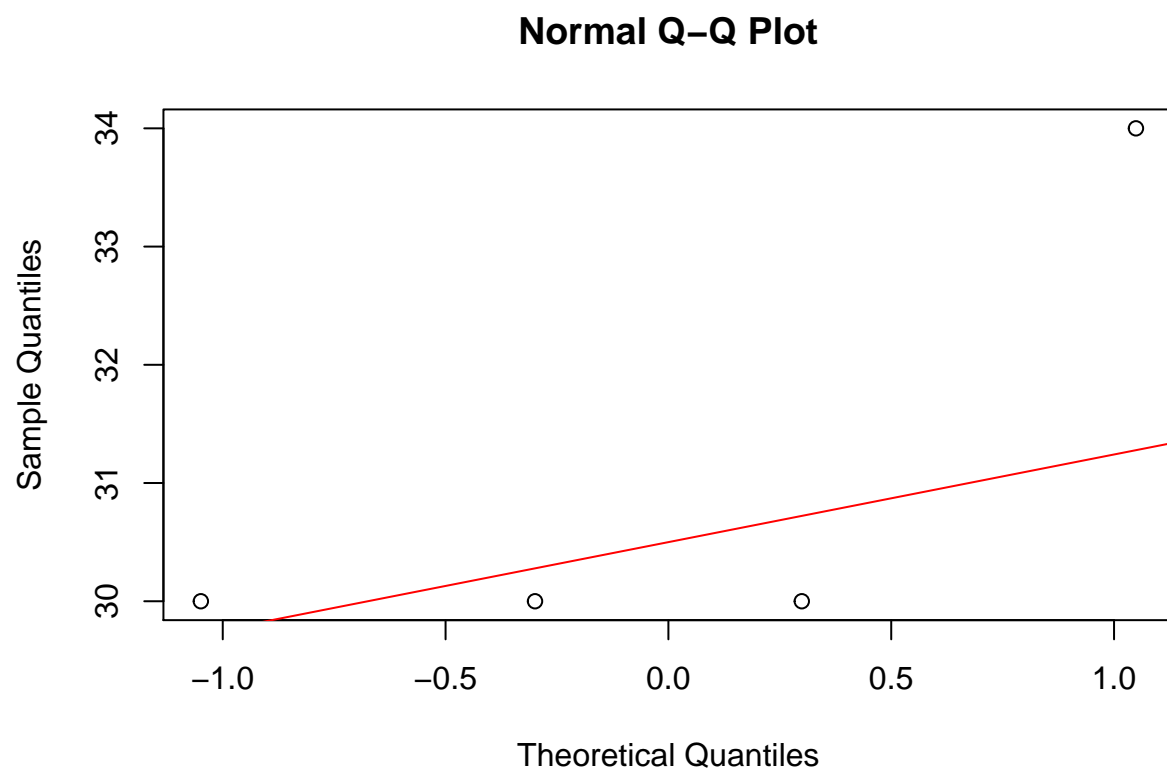


Normal Q-Q Plot



Normal Q-Q Plot



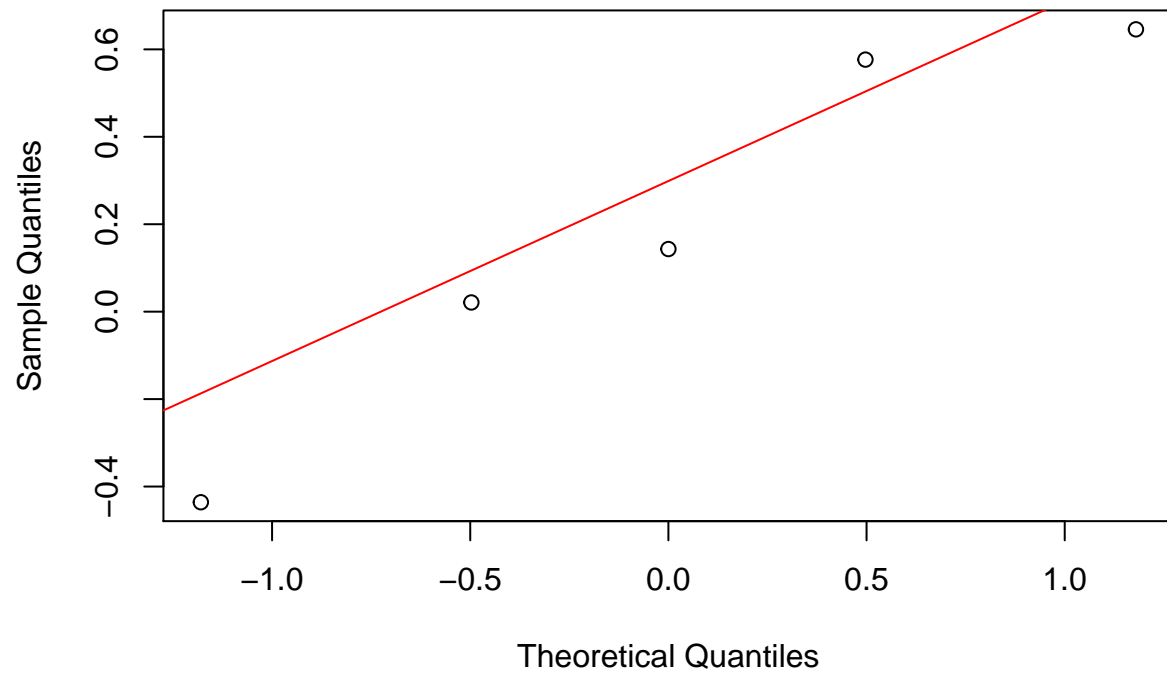


F test and randomization test for equal variance of number of cluster between treatments for each block in chardonnay 2019

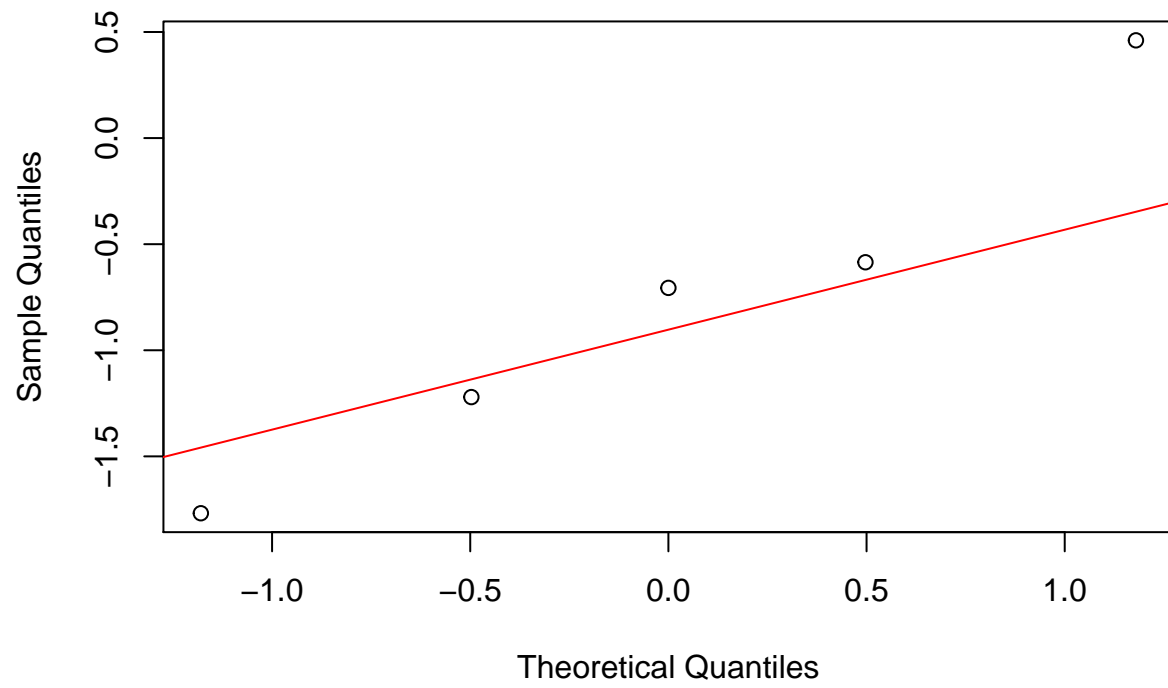
```
## [1] "F test"
## [1] 0.6842653
## [1] "rand test"
## [1] 0.6428571
## [1] "F test"
## [1] 0.7860631
## [1] "rand test"
## [1] 0.7571429
## [1] "F test"
## [1] 0.2464142
## [1] "rand test"
## [1] 0.1714286
## [1] "F test"
## [1] 0.5777467
## [1] "rand test"
## [1] 0.6
## [1] "F test"
## [1] 0.009387336
## [1] "rand test"
## [1] 0.04285714
```

QQ plot for the growth PC in merlot 2020 (for each block/treatment combo)

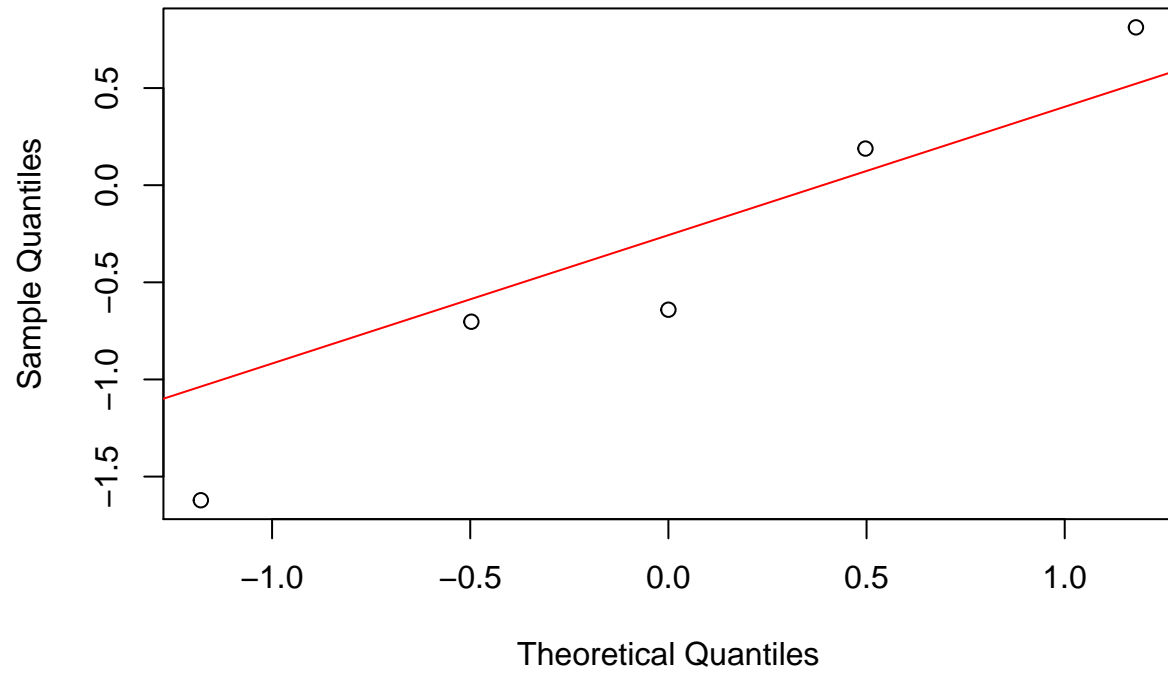
Normal Q-Q Plot



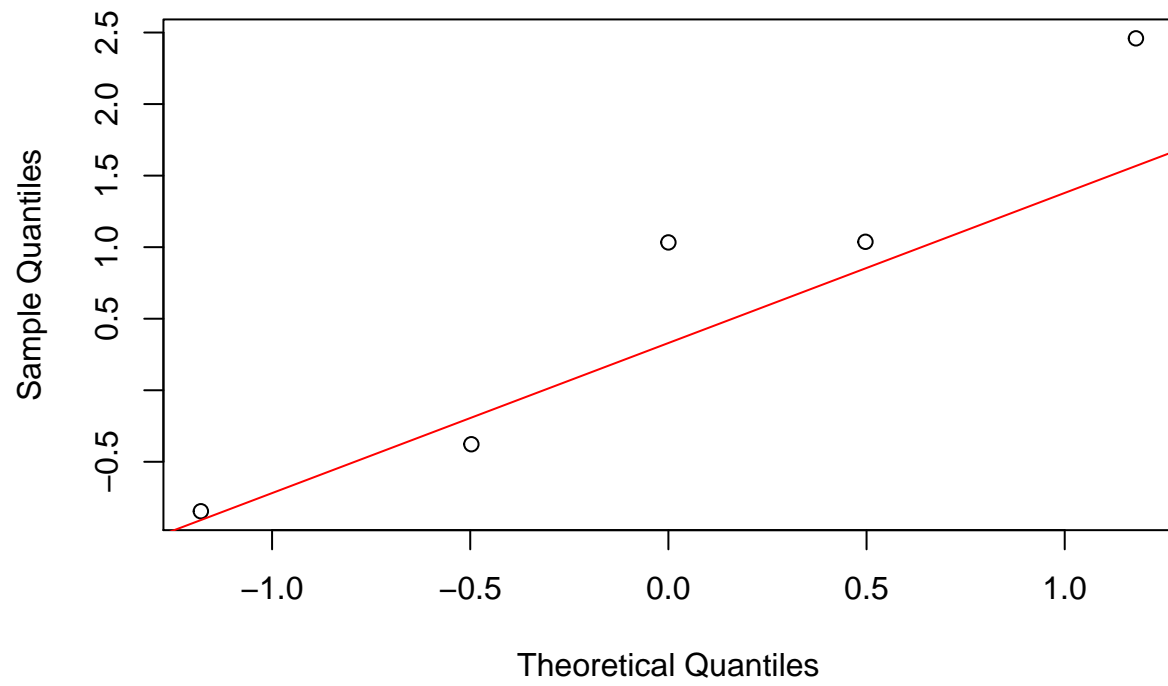
Normal Q-Q Plot



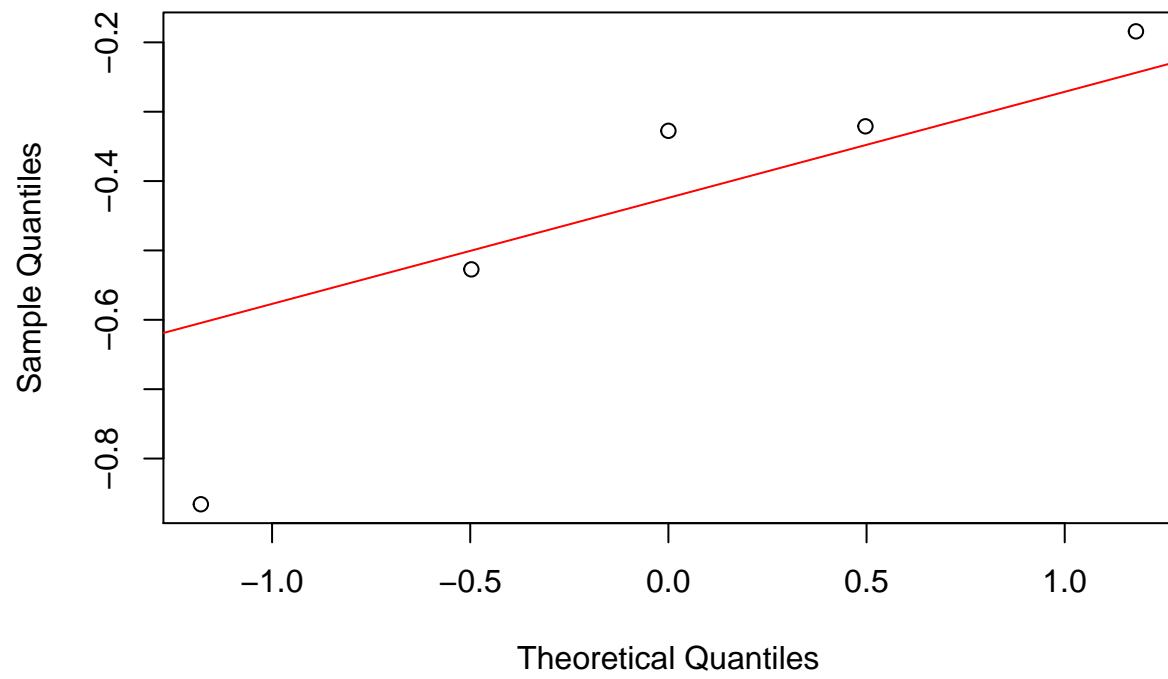
Normal Q-Q Plot



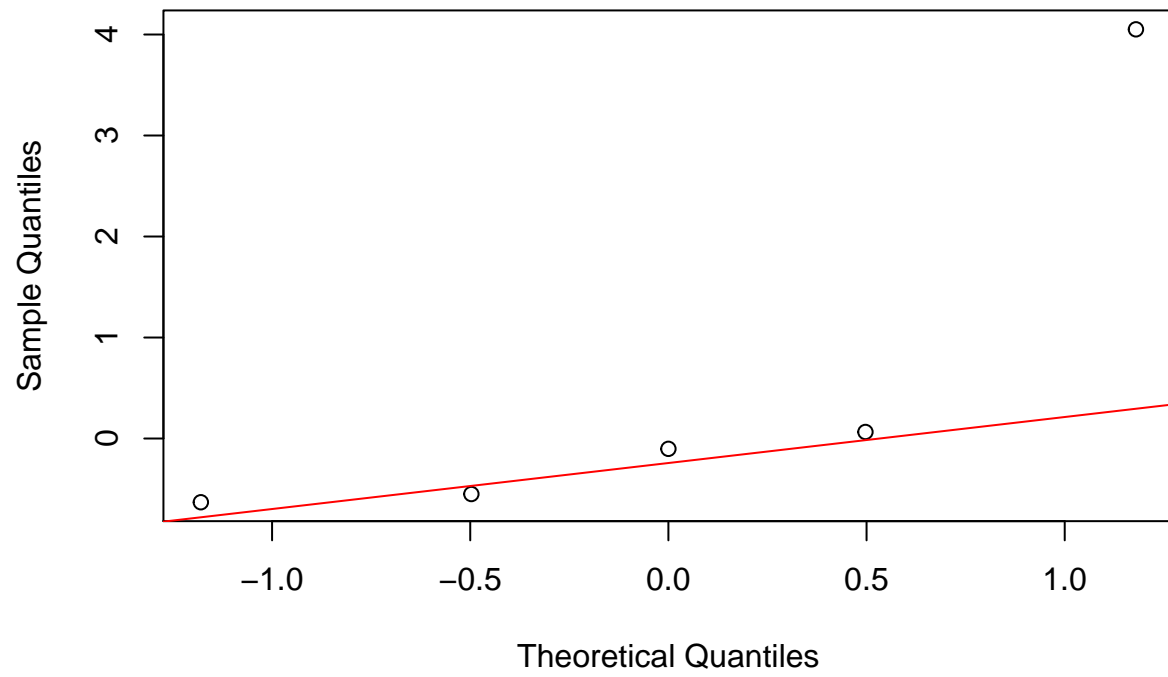
Normal Q-Q Plot



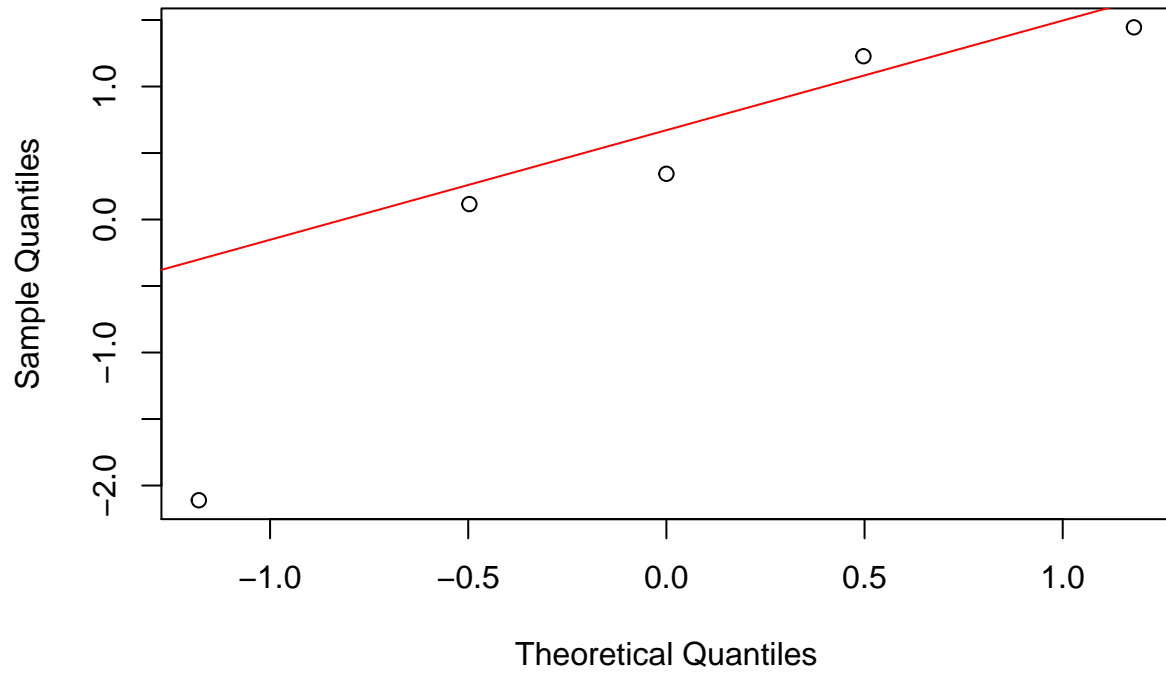
Normal Q-Q Plot



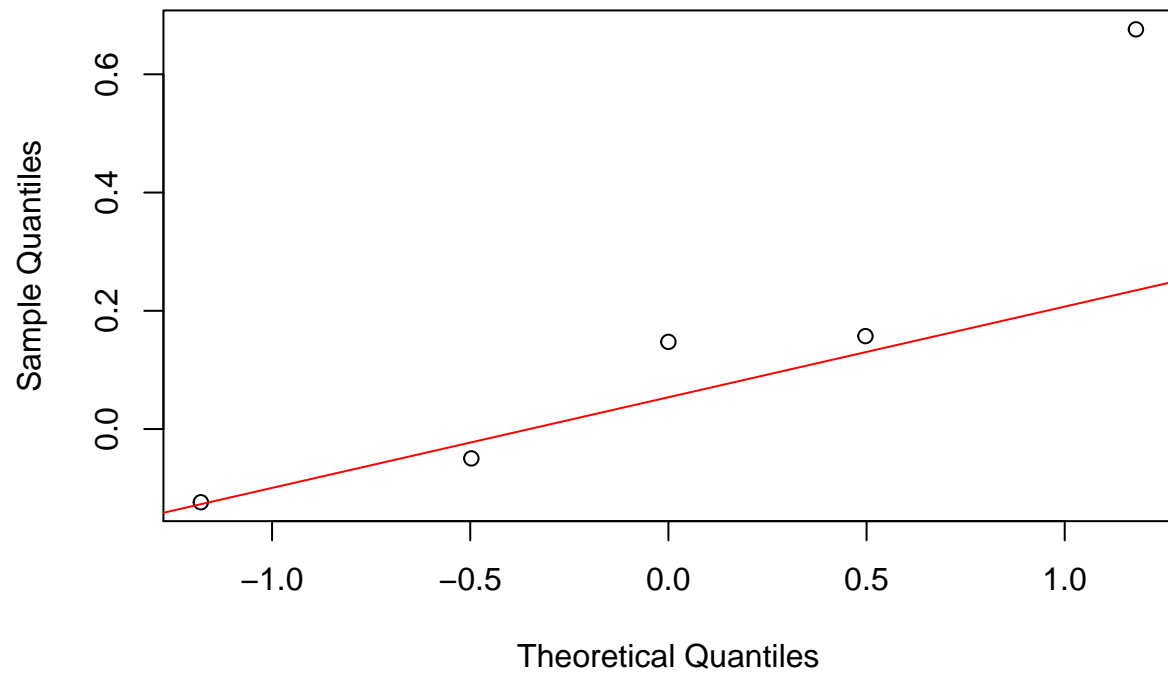
Normal Q-Q Plot



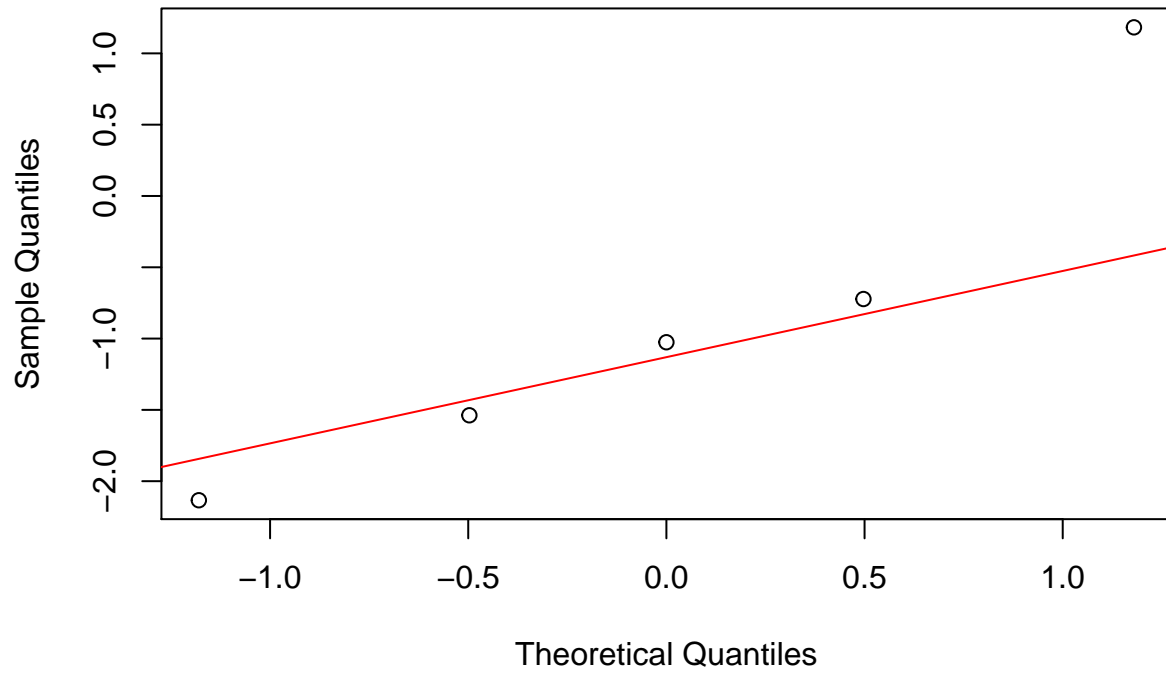
Normal Q-Q Plot



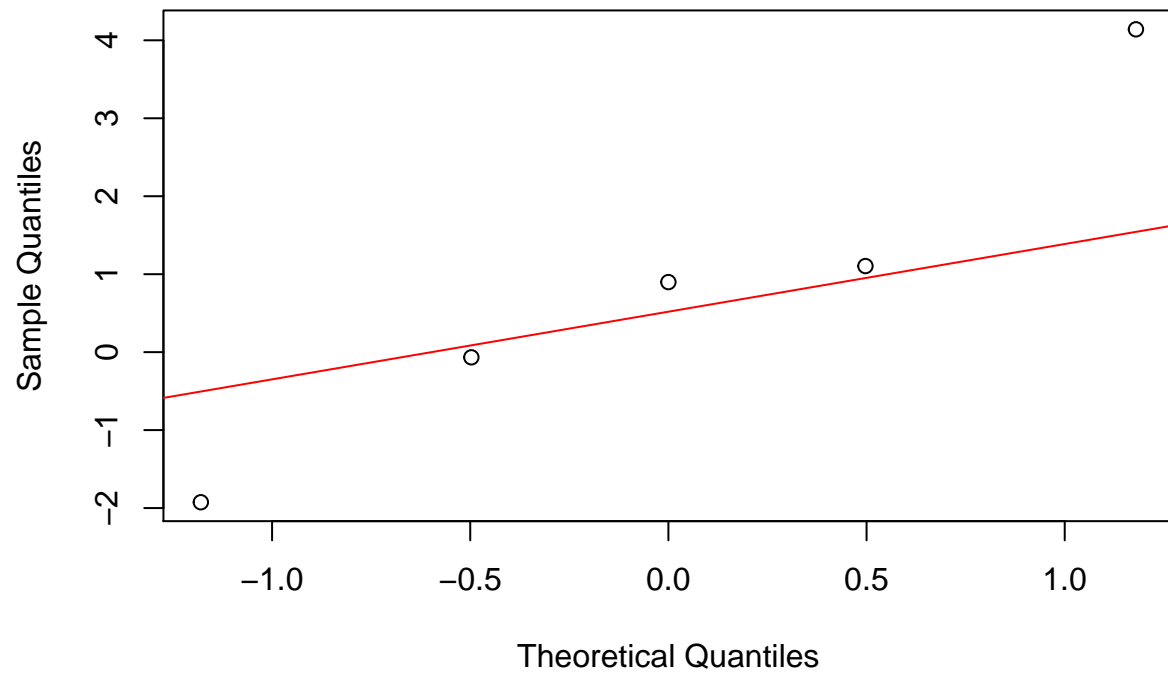
Normal Q-Q Plot



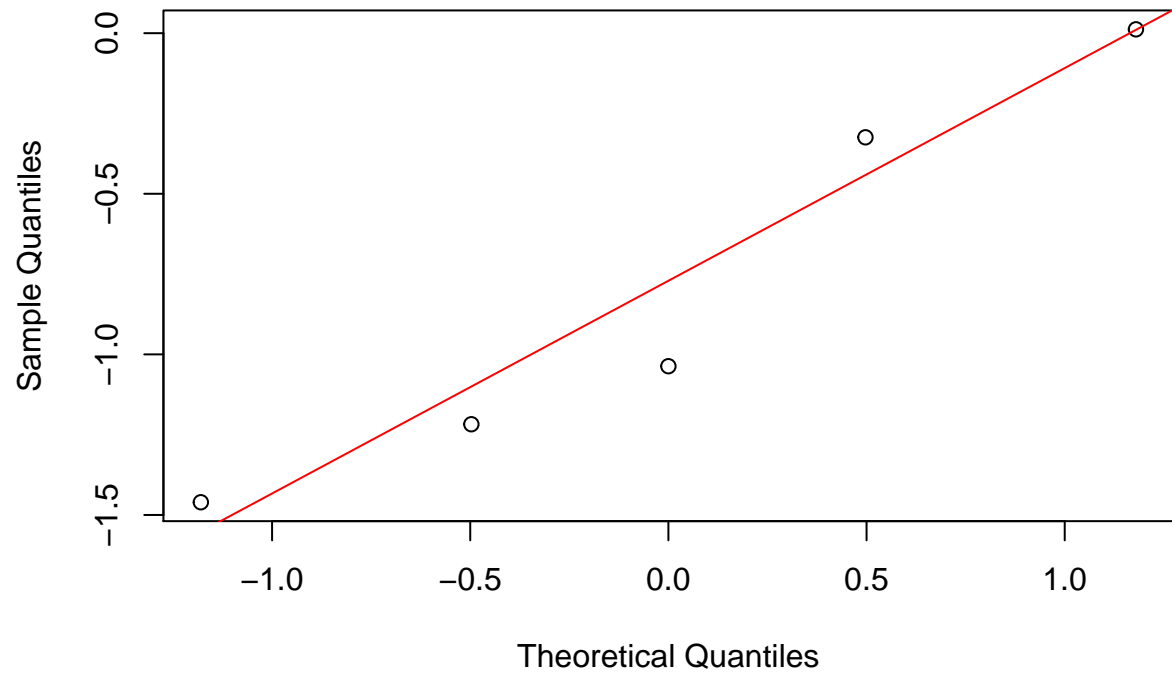
Normal Q-Q Plot

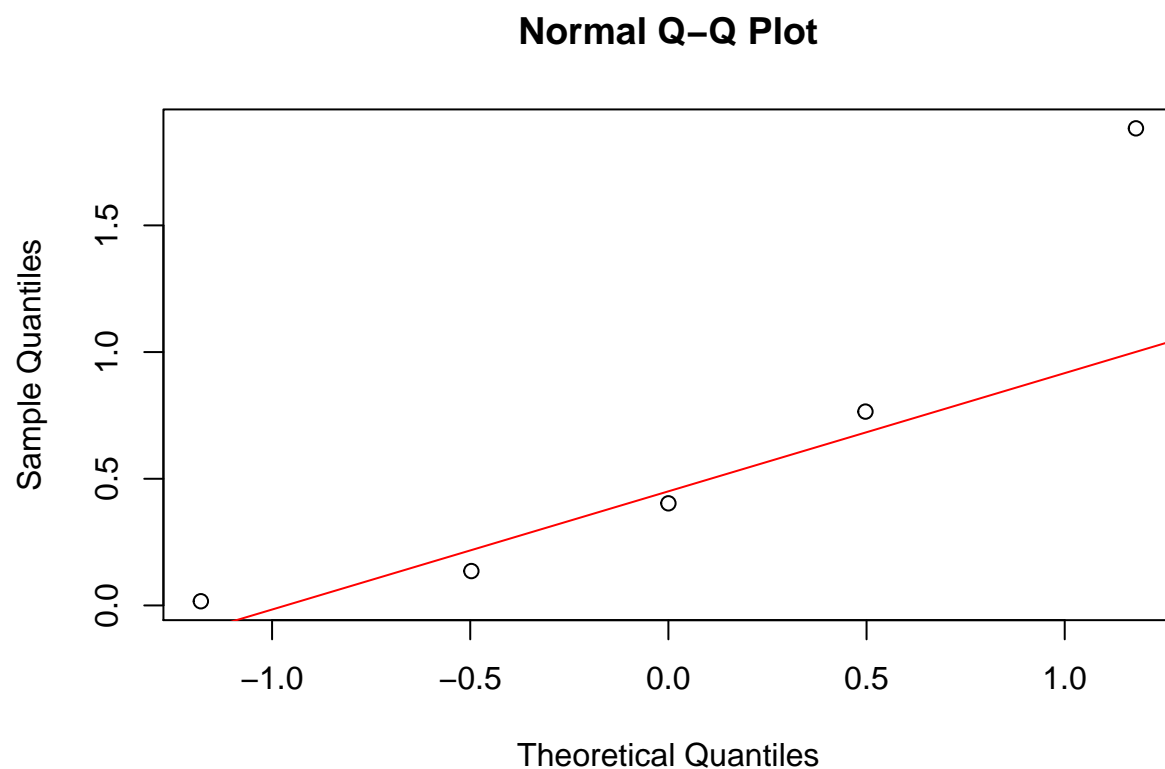


Normal Q-Q Plot



Normal Q-Q Plot





F test and randomization test for equal variance between treatments for each block in merlot 2020

```
## [1] "F test"
## [1] 0.2494025
## [1] "rand test"
## [1] 0.4722222
## [1] "F test"
## [1] 0.5232279
## [1] "rand test"
## [1] 0.5119048
## [1] "F test"
## [1] 0.001876501
## [1] "rand test"
## [1] 0.6706349
## [1] "F test"
## [1] 0.01270178
## [1] "rand test"
## [1] 0.1230159
## [1] "F test"
## [1] 0.3007981
## [1] "rand test"
## [1] 0.6944444
## [1] "F test"
## [1] 0.7247435
## [1] "rand test"
## [1] 0.9087302
```

We see that the normality assumptions are slightly violated, and the equal variance assumptions are also violated even though we only compared variances between treatments within each block. Therefore we will use both a parametric and a non parametric test in the following analysis. (Should repeat the above for all other variables.)

Univariate analysis

Univariate boxplots

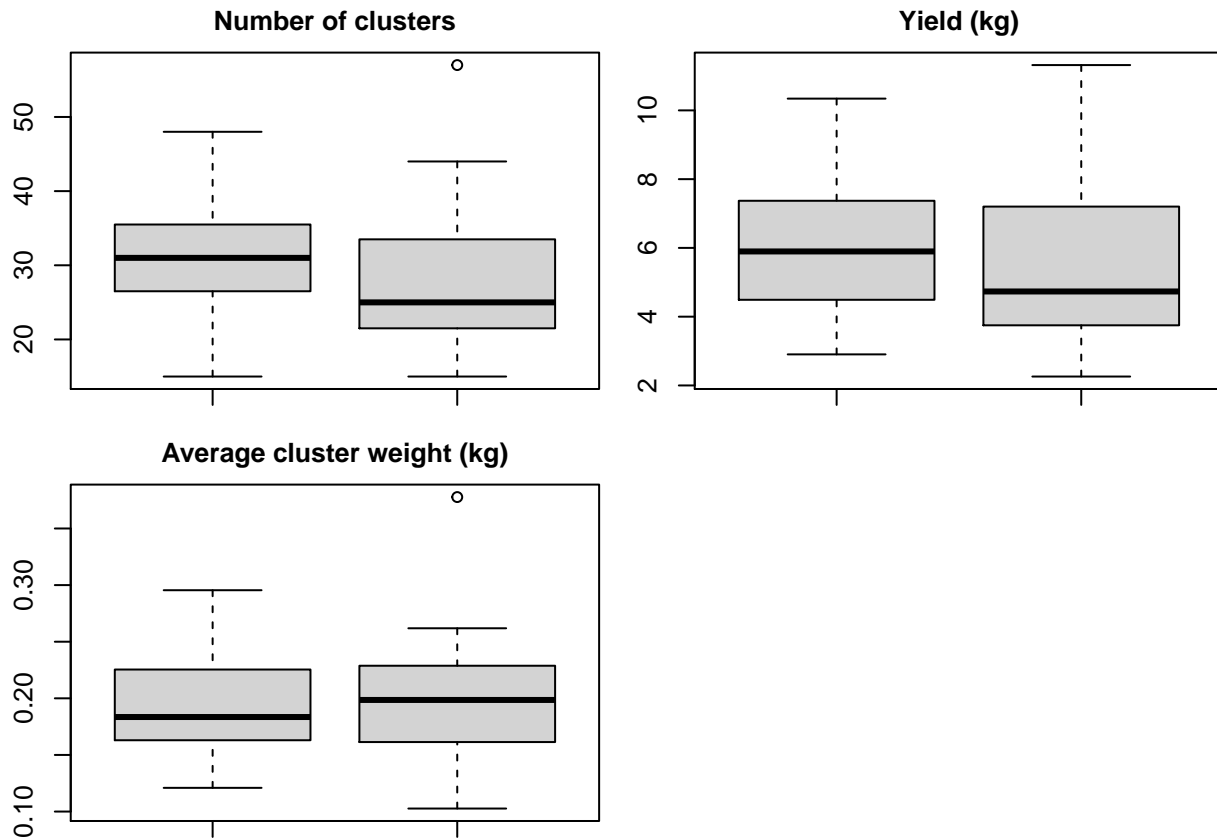
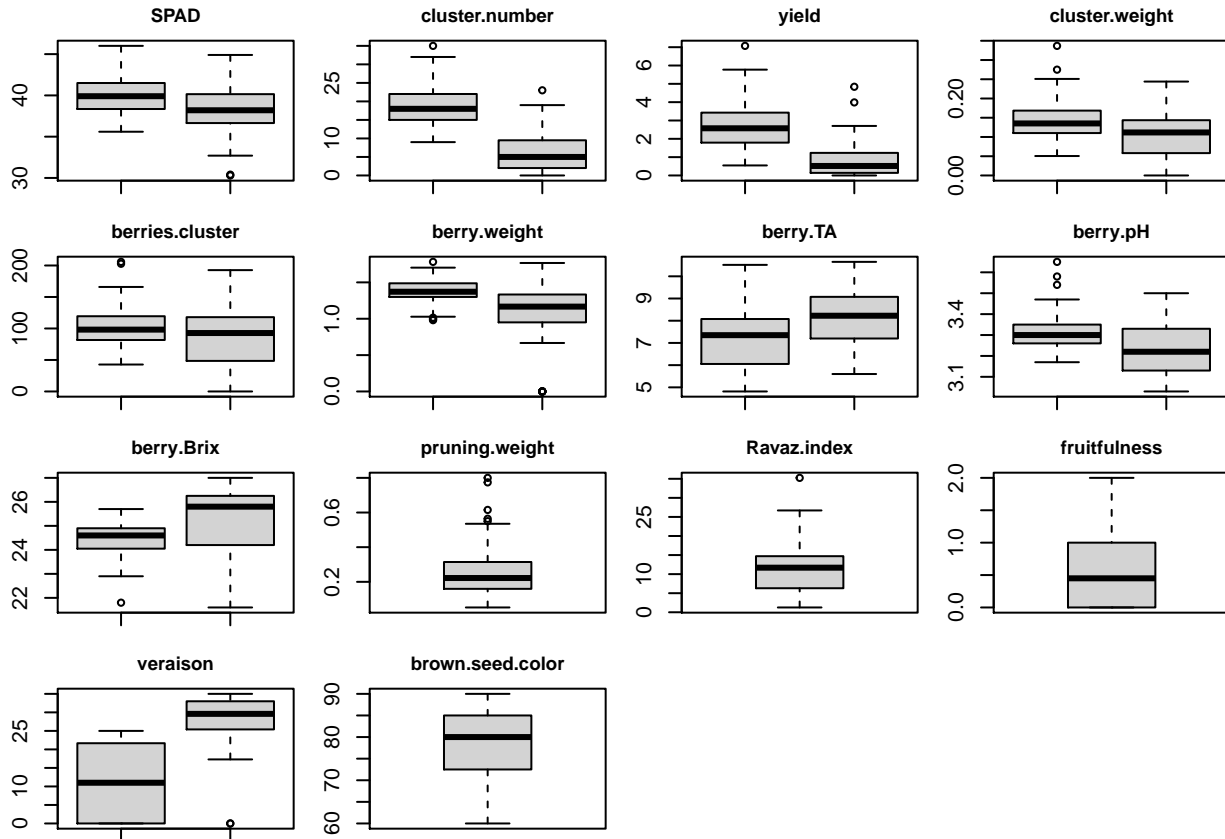


Figure 1: Boxplots of the overall distributions of factors looked at for Chardonnay, divided by year. These plots give us an idea of if there are any glaring outliers.



Checking assumptions for statistical tests

We will be running ANOVAs on all outcome variables except Bloom and Veraison to determine whether the treatment influenced the outcome variables. ANOVA tests have two main assumptions:

1. Homoscedasticity: each combination of the treatment and blocking factors have equal variances.
2. Normality: the outcome variable is approximately normal for each combination of the treatment and blocking factors.
3. Independence: the samples tested are obtained independently.
4. Equal sample size: each combination of the treatment and blocking factors have equal sample sizes.

Homoscedasticity



```
## [[1]]
## with.mids(data = merlot2019mice, expr = boxplot(cluster.weight ~
##   paste0(block, treatment)))
##
## [[2]]
## with.mids(data = merlot2019mice, expr = boxplot(cluster.weight ~
##   paste0(block, treatment)))
##
## [[3]]
## with.mids(data = merlot2019mice, expr = boxplot(cluster.weight ~
##   paste0(block, treatment)))
```

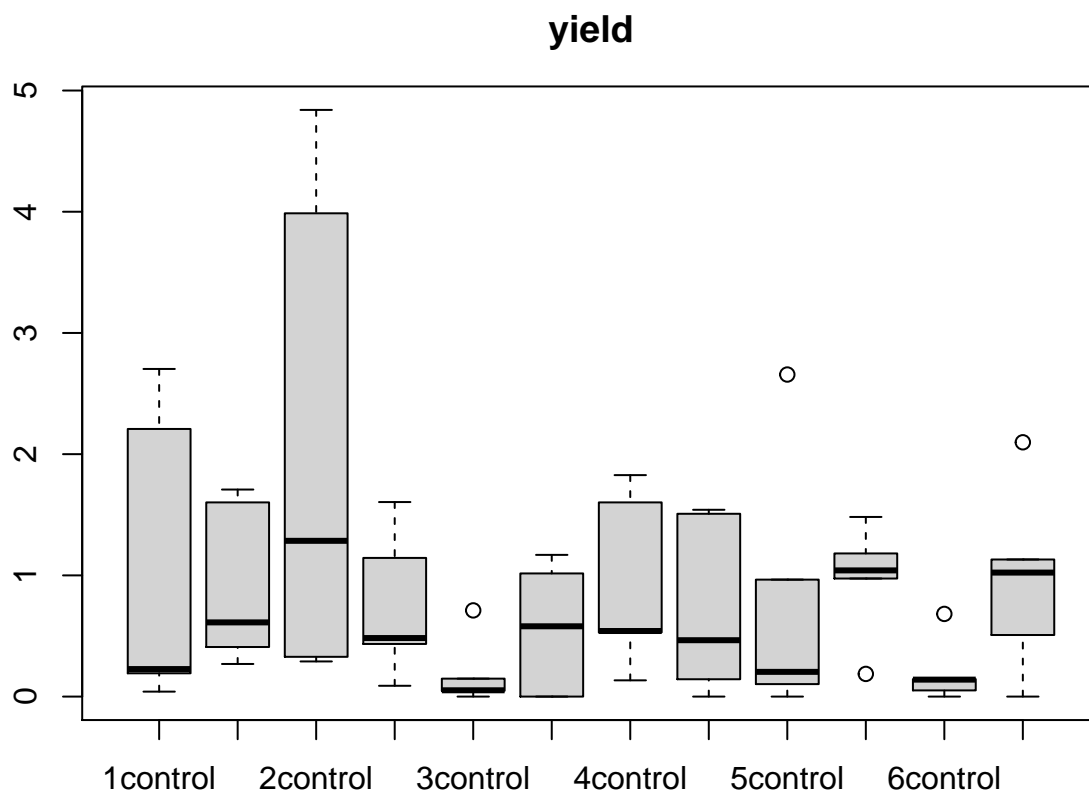
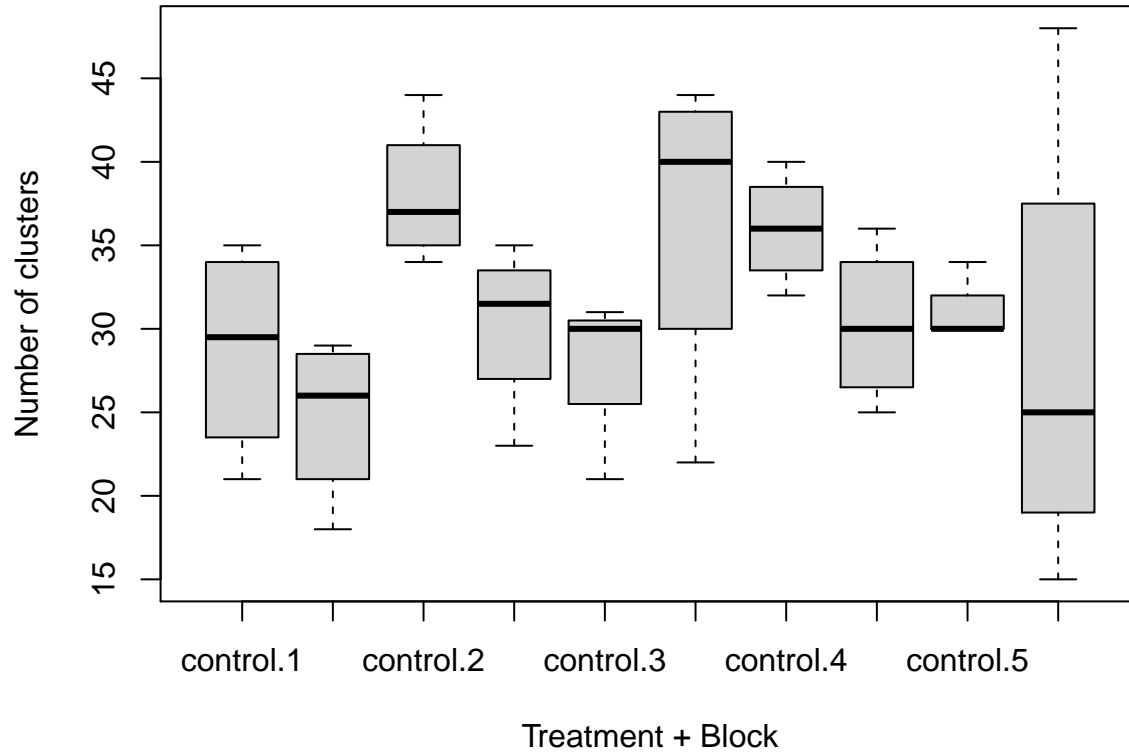


Figure 2: Distributions of variables for which equal variance assumptions were violated for at least one of the subgroups being analyzed.



Normality

We used Shapiro-Wilkes tests to check the normality assumption of the ANOVA tests, checking each block-treatment combination for each of the tested variables. For any Shapiro-Wilkes tests that resulted in a potentially significant breach from normality, the Q-Q plots were plotted (Figures 3, 4, 6, ??).

Upon observations of the Q-Q plots, it looks like deviations are largely due to outliers. This may pose a problem in the accuracy of the ANOVA tests as ANOVA can be sensitive to outliers.

Statistical analyses

Cox PH models for censored data

As previously discussed, Bloom and Veraison were both censored data for the Merlot datasets. For some of the plants, 100% bloom or veraison had already occurred, so the event (50% bloom or veraison) was left-censored. For other plants, the berries or flowers had not yet grown, indicating right-censoring (the event had not yet been observed).

Both bloom and veraison were measured three times (at times T1, T2, and T3, let's say), and the approximate event date (50% veraison or 50% bloom) was determined using a linear regression. There are two types of missing data:

- (1) The event did not occur because no berries or flowers grew. This is known as right-censored data. In this case, the event happened in the interval $(T3, \text{Inf})$
- (2) The event already occurred prior to the first observation. This is known as left-censored data. In this case, the event happened sometime in the interval $(-\text{Inf}, T1]$.

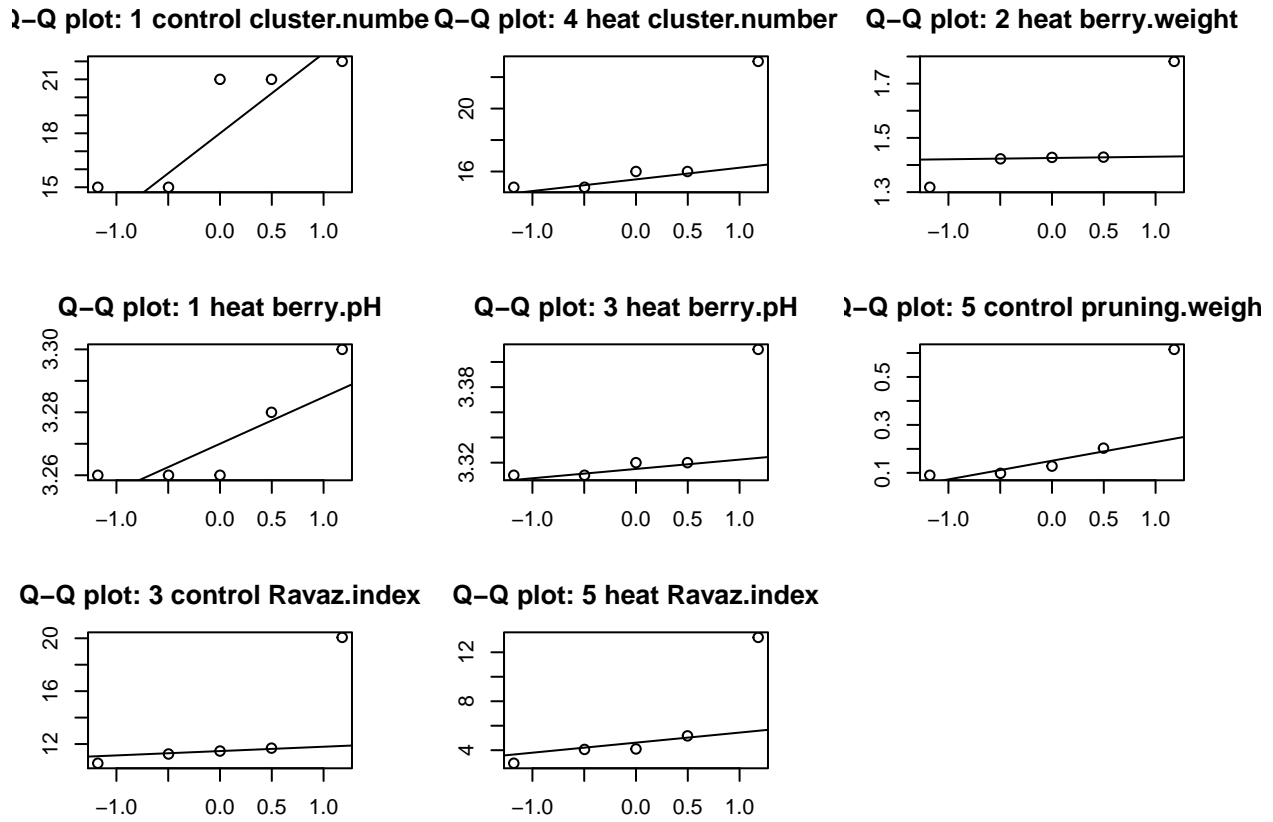


Figure 3: Q-Q plots for treatment-block groups in the Merlot 2019 data that indicated a potentially significant breach from normality. The number in the title for each plot indicates the block, followed by the treatment and the variable in question.

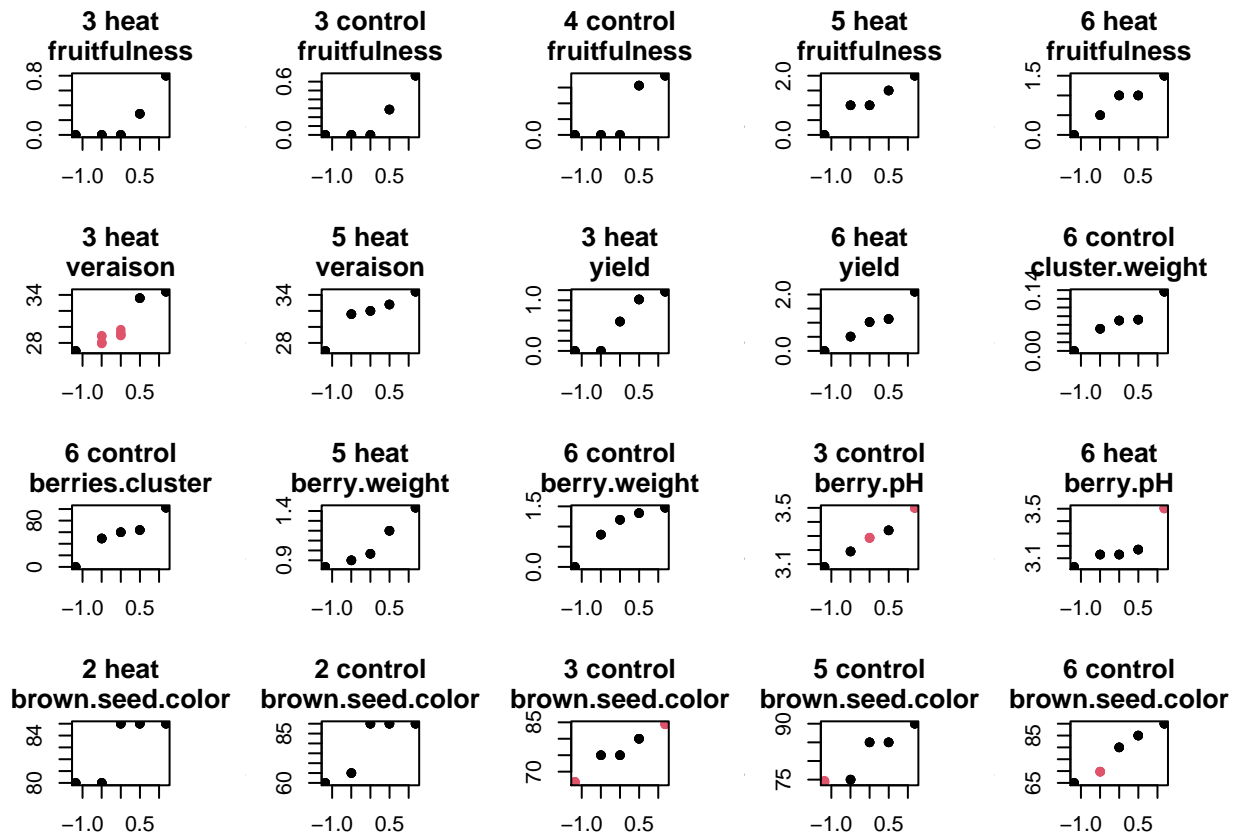


Figure 4: Q-Q plots for treatment-block groups in the Merlot 2020 data that indicated a potentially significant breach from normality. The number in the title for each plot indicates the block, followed by the treatment and the variable in question. Imputed data points are plotted in red.

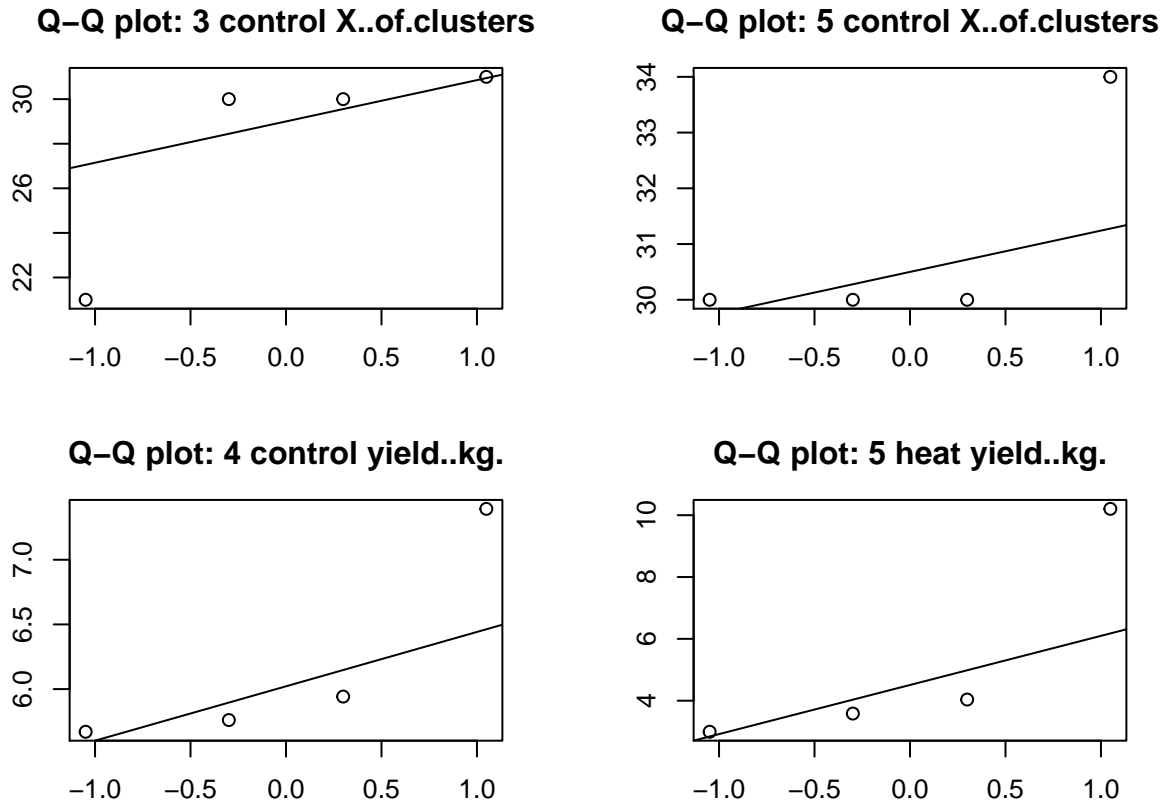


Figure 5: Q-Q plots for treatment-block groups in the Chardonnay 2019 data that indicated a potentially significant breach from normality. The number in the title for each plot indicates the block, followed by the treatment and the variable in question.

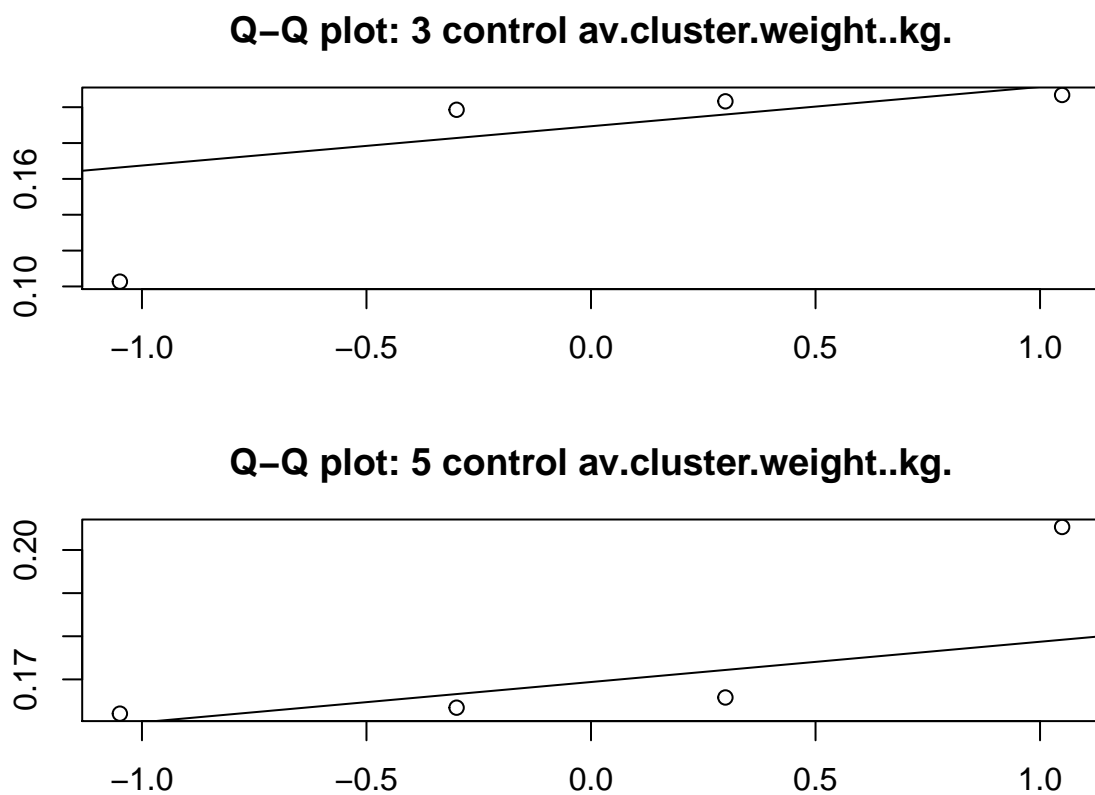


Figure 6: Q-Q plots for treatment-block groups in the Chardonnay 2019 data that indicated a potentially significant breach from normality. The number in the title for each plot indicates the block, followed by the treatment and the variable in question.

We have been in contact with the client to try to determine the exact T1 and T3 for each of the dates. However, in the absence of this data, we can use Multiple Imputation as our T1 and T3 dates. Additionally, for veraison, until we hear from the client, we can impute the lower bound of the interval for left-censored data as the 50% bloom date.

After treating the data as interval data, we can then construct Cox proportional hazards models to fit the data. We used tools in an R package called `icenReg`, which uses bootstrapping for inference on the regression parameters to perform hypothesis testing.

Results indicate that neither the treatment nor the block effects influenced Veraison in either 2019 or 2020. Bloom was influenced by block effect but not by the treatment.

Aligned rank transform ANOVA (nonparametric test for two-way layout) and ANOVA test

Chardonnay 2019 number of clusters

```
## Warning: package 'ARTool' was built under R version 4.0.3
## Registered S3 methods overwritten by 'lme4':
##   method                      from
##   cooks.distance.influence.merMod car
##   influence.merMod              car
##   dfbeta.influence.merMod       car
##   dfbetas.influence.merMod      car
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Analysis of Variance Table (Type I)
## Model: No Repeated Measures (lm)
## Response: art(X..of.clusters)
##
##              Df Df.res F value  Pr(>F)
## 1 treatment      1      30  1.2226 0.277632
## 2 block          4      30  2.4189 0.070334 .
## 3 treatment:block 4      30  2.4056 0.071534 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              Df Sum Sq Mean Sq F value Pr(>F)
## treatment      1   55.2   55.22   1.176  0.287
## block          4  285.2   71.29   1.518  0.222
## treatment:block 4  322.6   80.66   1.718  0.172
## Residuals     30 1408.8   46.96
```

Chardonnay 2019 yield

```
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Analysis of Variance Table (Type I)
## Model: No Repeated Measures (lm)
## Response: art(yield..kg.)
##
##              Df Df.res  F value  Pr(>F)
## 1 treatment      1      30 0.991545 0.32733
## 2 block          4      30 1.355386 0.27276
## 3 treatment:block 4      30 0.046131 0.99576
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##           Df Sum Sq Mean Sq F value Pr(>F)
## treatment    1   1.45   1.452   0.373  0.546
## block         4  20.44   5.111   1.312  0.288
## treatment:block 4   0.03   0.007   0.002  1.000
## Residuals    30 116.85   3.895
```

Chardonnay 2019 average cluster weight

```
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Analysis of Variance Table (Type I)
## Model: No Repeated Measures (lm)
## Response: art(av.cluster.weight..kg.)
##
##           Df Df.res F value  Pr(>F)
## 1 treatment    1     30 0.00902 0.92497
## 2 block         4     30 0.64523 0.63454
## 3 treatment:block 4     30 1.36546 0.26932
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##           Df Sum Sq Mean Sq F value Pr(>F)
## treatment    1 0.00001 0.0000057  0.003  0.956
## block         4 0.00716 0.0017912  0.964  0.441
## treatment:block 4 0.01036 0.0025897  1.394  0.260
## Residuals    30 0.05572 0.0018573
```

Chardonnay 2020 number of clusters

```
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Analysis of Variance Table (Type I)
## Model: No Repeated Measures (lm)
## Response: art(X..of.clusters)
##
##           Df Df.res F value  Pr(>F)
## 1 treatment    1     30  1.0012 0.325025
## 2 block         4     30  2.9615 0.035634 *
## 3 treatment:block 4     30  2.8050 0.043283 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##           Df Sum Sq Mean Sq F value Pr(>F)
## treatment    1   72.9   72.90   1.181 0.2859
## block         4  690.6  172.65   2.796 0.0438 *
## treatment:block 4  499.6  124.90   2.023 0.1165
## Residuals    30 1852.5   61.75
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

block factor was found significant to the number of clusters in both tests.

Chardonnay 2020 yield

```
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Analysis of Variance Table (Type I)
## Model: No Repeated Measures (lm)
## Response: art(yield..kg.)
##
##           Df Df.res F value  Pr(>F)
## 1 treatment      1      30 0.11283 0.739286
## 2 block          4      30 2.99336 0.034257 *
## 3 treatment:block 4      30 1.50554 0.225576
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##           Df Sum Sq Mean Sq F value Pr(>F)
## treatment      1    0.54    0.541    0.133 0.7182
## block          4   53.14   13.285    3.256 0.0248 *
## treatment:block 4   20.75    5.188    1.271 0.3031
## Residuals      30  122.41    4.080
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Block factor was found significant to yield in both tests.

Chardonnay 2020 average cluster weight

```
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Analysis of Variance Table (Type I)
## Model: No Repeated Measures (lm)
## Response: art(av.cluster.weight..kg.)
##
##           Df Df.res F value  Pr(>F)
## 1 treatment      1      30 0.71637 0.404037
## 2 block          4      30 2.45415 0.067265 .
## 3 treatment:block 4      30 0.36349 0.832608
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##           Df Sum Sq Mean Sq F value Pr(>F)
## treatment      1 0.00190 0.001902    0.824 0.3712
## block          4 0.02151 0.005377    2.329 0.0788 .
## treatment:block 4 0.00310 0.000775    0.336 0.8516
## Residuals      30 0.06925 0.002308
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Merlot 2019 growth PC

```
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Analysis of Variance Table (Type I)
## Model: No Repeated Measures (lm)
## Response: art(merlot2019.growth.pc)
##
##           Df Df.res F value  Pr(>F)
```

```
## 1 treatment      1      48  2.8169  0.099774  .
## 2 block          5      48  7.0881  4.9361e-05 ***
## 3 treatment:block 5      48  1.6448  0.166313
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##              Df Sum Sq Mean Sq F value    Pr(>F)
## treatment      1   3.44   3.436    4.824 0.03294 *
## block          5  23.13   4.626    6.495 0.00011 ***
## treatment:block 5   6.58   1.315    1.846 0.12166
## Residuals     48  34.19   0.712
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Again, block was found significant to the growth of grape (specifically what?) in both test. Treatment was also found significant in ANOVA, but ANOVA assumption does not hold, the result may not be reliable.

Merlot 2019 yield PC

```
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Analysis of Variance Table (Type I)
## Model: No Repeated Measures (lm)
## Response: art(merlot2019.yield.pc)
##
##              Df Df.res F value    Pr(>F)
## 1 treatment      1      48  1.5771 0.2152602
## 2 block          5      48  4.8827 0.0010855 **
## 3 treatment:block 5      48  1.8329 0.1242053
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##              Df Sum Sq Mean Sq F value    Pr(>F)
## treatment      1   4.31   4.310    2.063 0.157359
## block          5  54.20  10.841    5.190 0.000692 ***
## treatment:block 5  21.60   4.321    2.069 0.085753 .
## Residuals     48 100.26   2.089
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Block was again found significantly influential to yield

Merlot 2019 quality PC

```
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Analysis of Variance Table (Type I)
## Model: No Repeated Measures (lm)
## Response: art(merlot2019.quality.pc)
##
##              Df Df.res F value    Pr(>F)
## 1 treatment      1      48  1.59616 0.212550
## 2 block          5      48  2.90081 0.022858 *
## 3 treatment:block 5      48  0.86846 0.509231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## treatment      1   3.07   3.073   2.123 0.15160
## block          5  26.36   5.273   3.643 0.00712 **
## treatment:block 5   5.57   1.114   0.769 0.57641
## Residuals     48  69.48   1.448
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Merlot 2019 pruning PC

```
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Analysis of Variance Table (Type I)
## Model: No Repeated Measures (lm)
## Response: art(merlot2019.pruning.pc)
##
##               Df Df.res F value Pr(>F)
## 1 treatment      1     48  1.0191 0.317784
## 2 block          5     48  2.0081 0.094339 .
## 3 treatment:block 5     48  4.0950 0.003546 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##               Df Sum Sq Mean Sq F value Pr(>F)
## treatment      1   0.77   0.775   0.621 0.43461
## block          5  11.76   2.353   1.885 0.11451
## treatment:block 5  22.30   4.460   3.573 0.00794 **
## Residuals     48  59.92   1.248
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interaction was found significant in pruning (pruning weight, Ravaz index)

Merlot 2020 growth PC

```
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Analysis of Variance Table (Type I)
## Model: No Repeated Measures (lm)
## Response: art(merlot2020.growth.pc)
##
##               Df Df.res F value Pr(>F)
## 1 treatment      1     48  5.62031 0.021817 *
## 2 block          5     48  0.63308 0.675400
## 3 treatment:block 5     48  2.98813 0.019904 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##               Df Sum Sq Mean Sq F value Pr(>F)
## treatment      1   7.33   7.326   5.205 0.027 *
## block          5   1.44   0.289   0.205 0.959
## treatment:block 5  12.56   2.512   1.784 0.134
## Residuals     48  67.57   1.408
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

treatment and interaction were found significantly related to the growth. Maybe a particular geographical area combined with the heating technique effect are beneficial for the growth of grapes. We may also conduct **multiple comparison**.

Merlot 2020 yield PC

```
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Analysis of Variance Table (Type I)
## Model: No Repeated Measures (lm)
## Response: art(merlot2020.yield.pc)
##
##           Df Df.res F value  Pr(>F)
## 1 treatment      1      48 0.14007 0.70986
## 2 block          5      48 1.42597 0.23200
## 3 treatment:block 5      48 1.03306 0.40908
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##           Df Sum Sq Mean Sq F value Pr(>F)
## treatment      1    0.25   0.248   0.074 0.7874
## block          5   34.96   6.992   2.076 0.0848 .
## treatment:block 5   20.65   4.130   1.226 0.3116
## Residuals     48  161.70   3.369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Merlot 2020 quality PC

```
## Analysis of Variance of Aligned Rank Transformed Data
##
## Table Type: Analysis of Variance Table (Type I)
## Model: No Repeated Measures (lm)
## Response: art(merlot2020.quality.pc)
##
##           Df Df.res F value  Pr(>F)
## 1 treatment      1      48 0.35140 0.55611
## 2 block          5      48 0.85977 0.51494
## 3 treatment:block 5      48 1.30212 0.27885
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##           Df Sum Sq Mean Sq F value Pr(>F)
## treatment      1    0.34   0.3423   0.135 0.715
## block          5    7.68   1.5362   0.605 0.697
## treatment:block 5   13.82   2.7641   1.088 0.379
## Residuals     48  121.95   2.5407
```

Only one treatment effect was found significant (growth in merlot 2020). Although some of the interaction effects are. Maybe heating only helps under certain geographical conditions?

Another thing to check is the actual estimate of the treatment/blocking/interaction effects. We can also construct SEs of each of these effects.

Also, we may perform **multiple comparison** to see exactly which level of block differs.

Can also combine data from both years and do repeated measures ANOVA?