Design and Results for Wine Tasting Experiment

```
## Get the data
data <- read.csv(file = "chehalem_winery.csv", header = T)
A <- factor(data$A, levels = c(-1,1), labels = c("Pommard", "Wadenswil"))
B <- factor(data$B, levels = c(-1,1), labels = c("Allier", "Troncais"))
C <- factor(data$C, levels = c(-1,1), labels = c("Old", "New"))
D <- factor(data$D, levels = c(-1,1), labels = c("Champagne", "Montrachet"))
E <- factor(data$E, levels = c(-1,1), labels = c("None", "All"))
F <- factor(data$F, levels = c(-1,1), labels = c("Light", "Medium"))
G <- factor(data$G, levels = c(-1,1), labels = c("None", "10%"))
H <- factor(data$H, levels = c(-1,1), labels = c("Low", "High"))
y <- data$y
```

Create a model with up to 2-factor interactions. Notice not everything was estimated due to aliasing.

```
library(FrF2)
model.2fi <- lm(y~(A+B+C+D+E+F+G+H)^2, data = data)
summary(model.2fi)
```

```
##
## Call:
## lm.default(formula = y ~ (A + B + C + D + E + F + G + H)^2, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -5.80  -1.25  -0.10   1.00   5.80
##
## Coefficients: (21 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.5000     0.2658  31.985  < 2e-16 ***
## A             0.8750     0.2658   3.293 0.001619 **
## B             0.9250     0.2658   3.481 0.000906 ***
## C             0.6250     0.2658   2.352 0.021772 *
## D            -2.3000     0.2658  -8.655 2.27e-12 ***
## E             1.1000     0.2658   4.139 0.000104 ***
## F            -1.0000     0.2658  -3.763 0.000367 ***
## G             1.5750     0.2658   5.927 1.35e-07 ***
## H            -0.3000     0.2658  -1.129 0.263168
## A:B          -0.3500     0.2658  -1.317 0.192532
## A:C           1.3000     0.2658   4.892 7.07e-06 ***
## A:D          -0.8750     0.2658  -3.293 0.001619 **
## A:E           0.4750     0.2658   1.787 0.078613 .
## A:F           0.3750     0.2658   1.411 0.163063
## A:G           0.4500     0.2658   1.693 0.095261 .
## A:H           1.2250     0.2658   4.610 1.98e-05 ***
## B:C               NA         NA      NA       NA
## B:D               NA         NA      NA       NA
## B:E               NA         NA      NA       NA
## B:F               NA         NA      NA       NA
## B:G               NA         NA      NA       NA
## B:H               NA         NA      NA       NA
## C:D               NA         NA      NA       NA
```

```
## C:E               NA         NA      NA       NA
## C:F               NA         NA      NA       NA
## C:G               NA         NA      NA       NA
## C:H               NA         NA      NA       NA
## D:E               NA         NA      NA       NA
## D:F               NA         NA      NA       NA
## D:G               NA         NA      NA       NA
## D:H               NA         NA      NA       NA
## E:F               NA         NA      NA       NA
## E:G               NA         NA      NA       NA
## E:H               NA         NA      NA       NA
## F:G               NA         NA      NA       NA
## F:H               NA         NA      NA       NA
## G:H               NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.377 on 64 degrees of freedom
## Multiple R-squared:  0.7873, Adjusted R-squared:  0.7374
## F-statistic: 15.79 on 15 and 64 DF,  p-value: 4.547e-16
```

```
aliases(model.2fi) # This gives us the aliasing structure
```

```
##
##  A:B = C:G = D:H = E:F
##  A:C = B:G = D:F = E:H
##  A:D = B:H = C:F = E:G
##  A:E = B:F = C:H = D:G
##  A:F = B:E = C:D = G:H
##  A:G = B:C = D:E = F:H
##  A:H = B:D = C:E = F:G
```

```
## Complete aliasing structure:
# use 8 here because k=8 is the number of factors being investigated
# and is the largest interaction possible
aliases(lm(y~(A+B+C+D+E+F+G+H)^8, data = data))
```

```
##
##  A = B:C:G = B:D:H = B:E:F = C:D:F = C:E:H = D:E:G = F:G:H = A:B:C:D:E = A:B:C:F:H = A:B:D:F:G = A:B
##  B = A:C:G = A:D:H = A:E:F = C:D:E = C:F:H = D:F:G = E:G:H = A:B:C:D:F = A:B:C:E:H = A:B:D:E:G = A:B
##  C = A:B:G = A:D:F = A:E:H = B:D:E = B:F:H = D:G:H = E:F:G = A:B:C:D:H = A:B:C:E:F = A:C:D:E:G = A:C
##  D = A:B:H = A:C:F = A:E:G = B:C:E = B:F:G = C:G:H = E:F:H = A:B:C:D:G = A:B:D:E:F = A:C:D:E:H = A:D
##  E = A:B:F = A:C:H = A:D:G = B:C:D = B:G:H = C:F:G = D:F:H = A:B:C:E:G = A:B:D:E:H = A:C:D:E:F = A:E
##  F = A:B:E = A:C:D = A:G:H = B:C:H = B:D:G = C:E:G = D:E:H = A:B:C:F:G = A:B:D:F:H = A:C:E:F:H = A:D
##  G = A:B:C = A:D:E = A:F:H = B:D:F = B:E:H = C:D:H = C:E:F = A:B:D:G:H = A:B:E:F:G = A:C:D:F:G = A:C
##  H = A:B:D = A:C:E = A:F:G = B:C:F = B:E:G = C:D:G = D:E:F = A:B:C:G:H = A:B:E:F:H = A:C:D:F:H = A:D
##  A:B = C:G = D:H = E:F = A:C:D:E = A:C:F:H = A:D:F:G = A:E:G:H = B:C:D:F = B:C:E:H = B:D:E:G = B:F:G
##  A:C = B:G = D:F = E:H = A:B:D:E = A:B:F:H = A:D:G:H = A:E:F:G = B:C:D:H = B:C:E:F = C:D:E:G = C:F:G
##  A:D = B:H = C:F = E:G = A:B:C:E = A:B:F:G = A:C:G:H = A:E:F:H = B:C:D:G = B:D:E:F = C:D:E:H = D:F:G
##  A:E = B:F = C:H = D:G = A:B:C:D = A:B:G:H = A:C:F:G = A:D:F:H = B:C:E:G = B:D:E:H = C:D:E:F = E:F:G
##  A:F = B:E = C:D = G:H = A:B:C:H = A:B:D:G = A:C:E:G = A:D:E:H = B:C:F:G = B:D:F:H = C:E:F:H = D:E:F
##  A:G = B:C = D:E = F:H = A:B:D:F = A:B:E:H = A:C:D:H = A:C:E:F = B:D:G:H = B:E:F:G = C:D:F:G = C:E:G
##  A:H = B:D = C:E = F:G = A:B:C:F = A:B:E:G = A:C:D:G = A:D:E:F = B:C:G:H = B:E:F:H = C:D:F:H = D:E:G
```

Identify the most influential factors. Note it appears as though factors: D, G, E, F have the largest main effects and the 2-factor interactions AC = DF, AH = FG, and AD = EG are most important.

```
effects <- 2*model.2fi$coefficients[2:length(model.2fi$coefficients)]
effects[order(abs(effects), decreasing = FALSE)]
```

```
##     H   A:B   A:F   A:G   A:E     C   A:D     A     B     F     E   A:H
## -0.60 -0.70  0.75  0.90  0.95  1.25 -1.75  1.75  1.85 -2.00  2.20  2.45
##   A:C     G     D   B:C   B:D   B:E   B:F   B:G   B:H   C:D   C:E   C:F
##  2.60  3.15 -4.60    NA    NA    NA    NA    NA    NA    NA    NA    NA
##   C:G   C:H   D:E   D:F   D:G   D:H   E:F   E:G   E:H   F:G   F:H   G:H
##    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
```

Let's try fitting a reduced model with just these terms.

```
model.red <- lm(y ~ A + B + C + D + E + F + G + D:F + F:G + E:G, data = data)
summary(model.red)
```

```
##
## Call:
## lm.default(formula = y ~ A + B + C + D + E + F + G + D:F + F:G +
##     E:G, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.500 -1.462 -0.175  1.087  6.800
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.5000     0.2772  30.666  < 2e-16 ***
## A             0.8750     0.2772   3.157 0.002366 **
## B             0.9250     0.2772   3.337 0.001366 **
## C             0.6250     0.2772   2.255 0.027320 *
## D            -2.3000     0.2772  -8.298 5.70e-12 ***
## E             1.1000     0.2772   3.969 0.000175 ***
## F            -1.0000     0.2772  -3.608 0.000580 ***
## G             1.5750     0.2772   5.682 2.92e-07 ***
## D:F           1.3000     0.2772   4.690 1.34e-05 ***
## F:G           1.2250     0.2772   4.419 3.59e-05 ***
## E:G          -0.8750     0.2772  -3.157 0.002366 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.479 on 69 degrees of freedom
## Multiple R-squared:  0.7505, Adjusted R-squared:  0.7144
## F-statistic: 20.76 on 10 and 69 DF,  p-value: < 2.2e-16
```

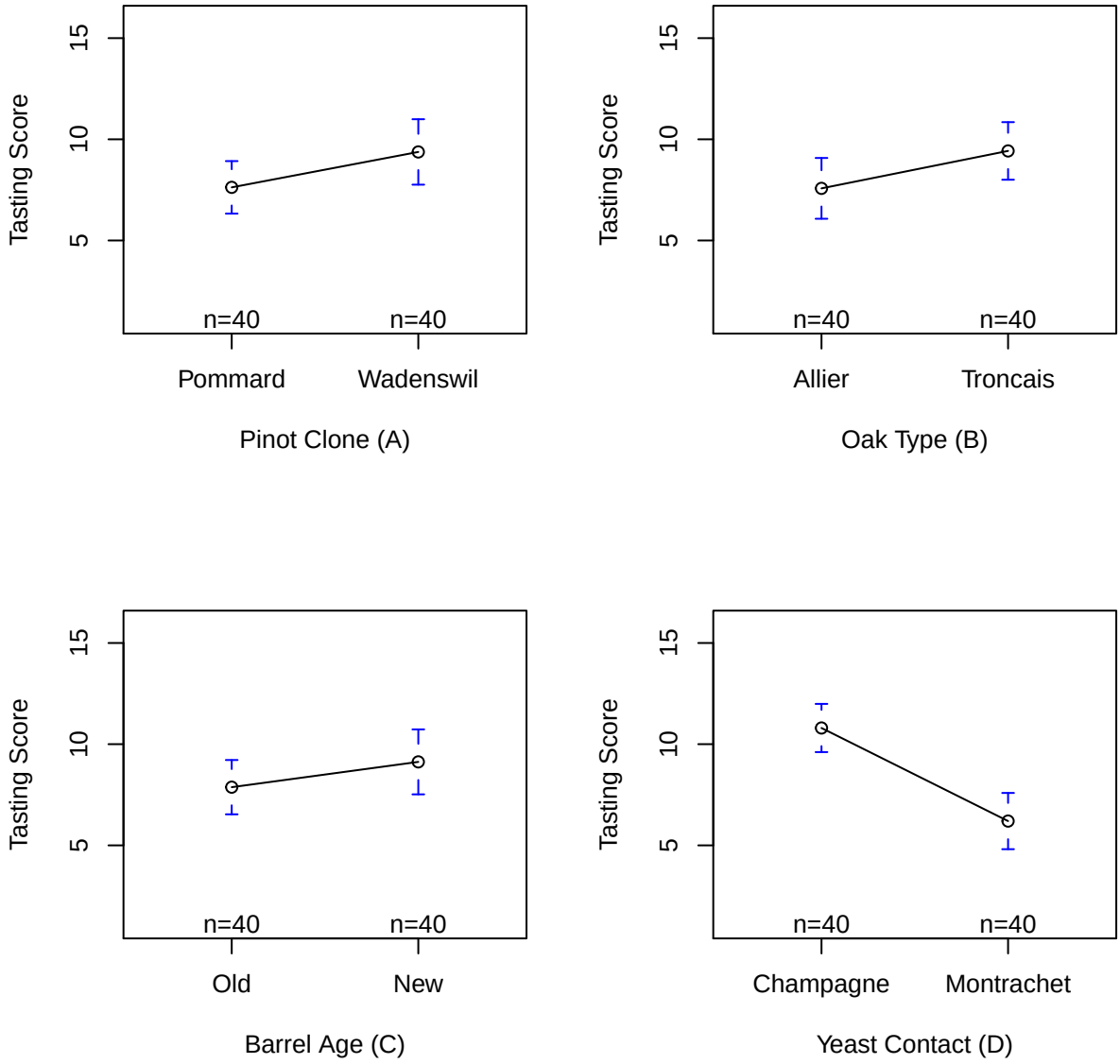
```
anova(model.red, model.2fi)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ A + B + C + D + E + F + G + D:F + F:G + E:G
## Model 2: y ~ (A + B + C + D + E + F + G + H)^2
##   Res.Df   RSS Df Sum of Sq      F  Pr(>F)
## 1     69 424.1
## 2     64 361.6  5      62.5 2.2124 0.06375 .
## ---
```

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

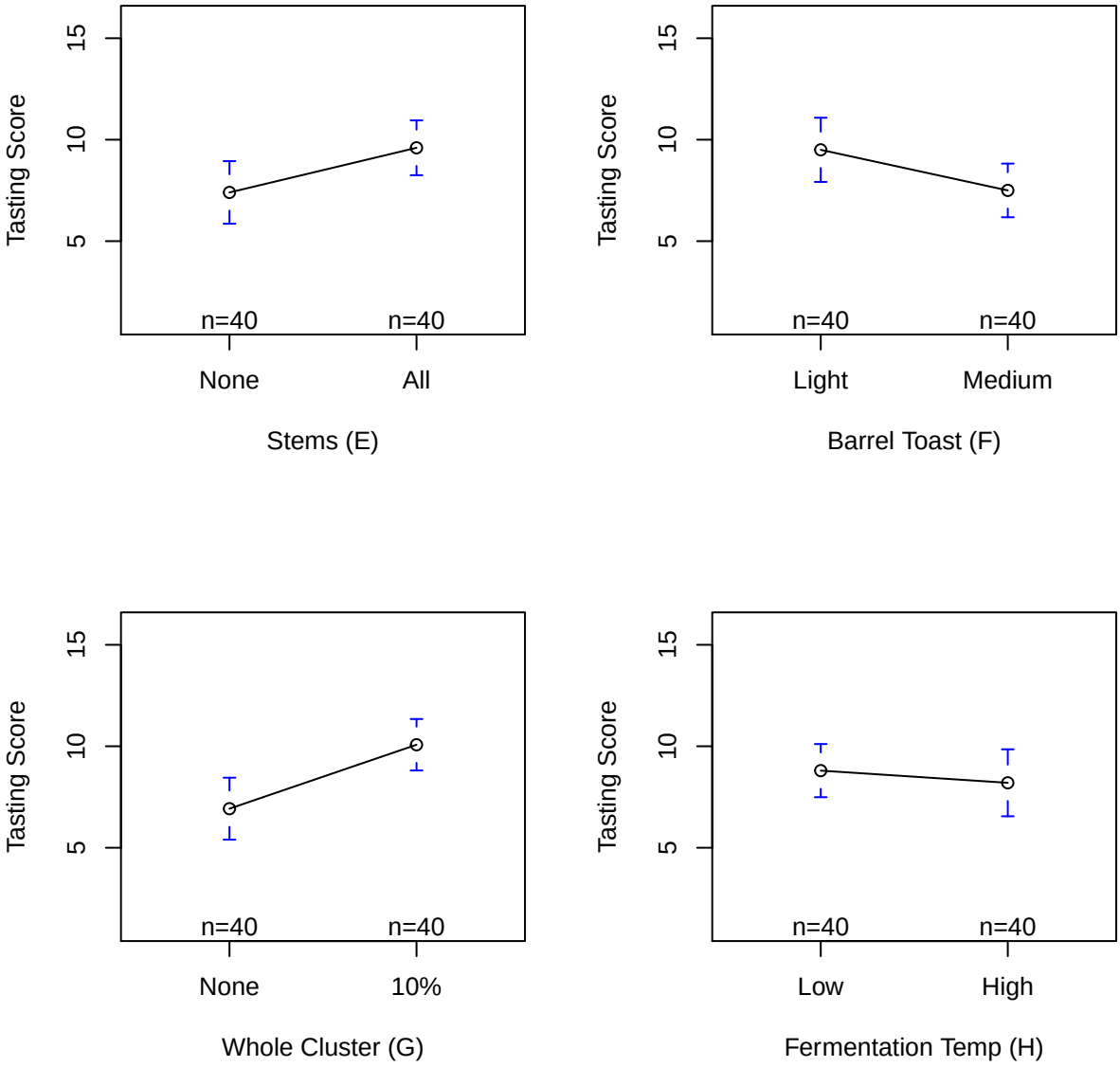
```r
## Main Effects plots
library(gplots)
par(mfrow=c(2,2), oma = c(0,0,2,0))
# par(mfrow=c(2,2), oma = c(0,0,0,0))
plotmeans(formula = y~A, ylab = "Tasting Score", xlab = "Pinot Clone (A)", ylim = c(1, 16), data = data
axis(side = 1, at = c(1,2), labels = c("Pommard", "Wadenswil"))
plotmeans(formula = y~B, ylab = "Tasting Score", xlab = "Oak Type (B)", ylim = c(1, 16), data = data, xa
axis(side = 1, at = c(1,2), labels = c("Allier", "Troncais"))
plotmeans(formula = y~C, ylab = "Tasting Score", xlab = "Barrel Age (C)", ylim = c(1, 16), data = data,
axis(side = 1, at = c(1,2), labels = c("Old", "New"))
plotmeans(formula = y~D, ylab = "Tasting Score", xlab = "Yeast Contact (D)", ylim = c(1, 16), data = da
axis(side = 1, at = c(1,2), labels = c("Champagne", "Montrachet"))
mtext("Main Effect Plots 1", outer = TRUE, cex = 1.5)
```

Main Effect Plots 1



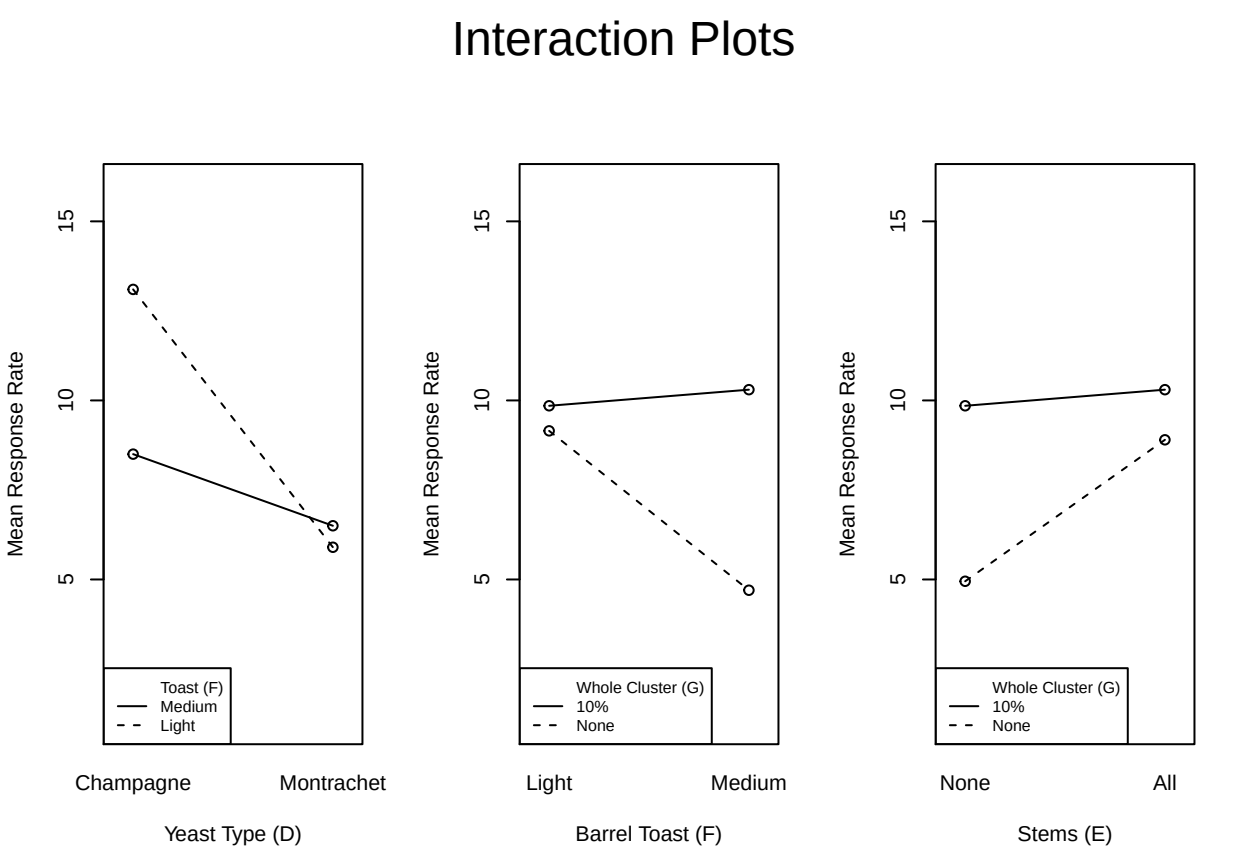

```
plotmeans(formula = y~E, ylab = "Tasting Score", xlab = "Stems (E)", ylim = c(1, 16), data = data, xaxt
axis(side = 1, at = c(1,2), labels = c("None", "All"))
plotmeans(formula = y~F, ylab = "Tasting Score", xlab = "Barrel Toast (F)", ylim = c(1, 16), data = data
axis(side = 1, at = c(1,2), labels = c("Light", "Medium"))
plotmeans(formula = y~G, ylab = "Tasting Score", xlab = "Whole Cluster (G)", ylim = c(1, 16), data = da
axis(side = 1, at = c(1,2), labels = c("None", "10%"))
plotmeans(formula = y~H, ylab = "Tasting Score", xlab = "Fermentation Temp (H)", ylim = c(1, 16), data =
axis(side = 1, at = c(1,2), labels = c("Low", "High"))
mtext("Main Effect Plots 2", outer = TRUE, cex = 1.5)
```

Main Effect Plots 2



Notes: It is clear that yeast type (D) and the amount of whole clusters (G) used during fermentation are most important, with no whole clusters and Montrachet yeast producing a better tasting wine. Although not significant, medium barrel toast (F) and no stems (E) seem to correspond to a better tasting Pinot Noir.

```r
## Interaction Plots
par(mfrow=c(1,3), oma = c(0,0,2,0))
# par(mfrow=c(1,3), oma = c(0,0,0,0))
interaction.plot(D, F, y, ylab = "Mean Response Rate", xlab = "Yeast Type (D)", main = "", ylim = c(1,
points(x = c(1,1), y = c(mean(data[data$D==-1 & data$F==-1,]$y),mean(data[data$D==-1 & data$F==1,]$y)),
points(x = c(2,2), y = c(mean(data[data$D==1 & data$F==-1,]$y),mean(data[data$D==1 & data$F==1,]$y)), p
legend("bottomleft", legend = c("Toast (F)","Medium", "Light"), lty = c(1,1,2), col=c("white", "black",
interaction.plot(F, G, y, ylab = "Mean Response Rate", xlab = "Barrel Toast (F)", main = "", ylim = c(1
points(x = c(1,1), y = c(mean(data[data$F==-1 & data$G==-1,]$y),mean(data[data$F==-1 & data$G==1,]$y)),
points(x = c(2,2), y = c(mean(data[data$F==1 & data$G==-1,]$y),mean(data[data$F==1 & data$G==1,]$y)), p
legend("bottomleft", legend = c("Whole Cluster (G)","10%", "None"), lty = c(1,1,2), col=c("white", "bla
interaction.plot(E, G, y, ylab = "Mean Response Rate", xlab = "Stems (E)", main = "", ylim = c(1, 16),
points(x = c(1,1), y = c(mean(data[data$E==-1 & data$G==-1,]$y),mean(data[data$E==-1 & data$G==1,]$y)),
points(x = c(2,2), y = c(mean(data[data$E==1 & data$G==-1,]$y),mean(data[data$E==1 & data$G==1,]$y)), p
legend("bottomleft", legend = c("Whole Cluster (G)","10%", "None"), lty = c(1,1,2), col=c("white", "bla
mtext("Interaction Plots", outer = TRUE, cex = 1.5)
```



Notes: If yeast type is Montrachet, barrel toast doesn't matter much, but if yeast type is Champagne, a medium barrell toast is best. And if barrel toast is chosen to be medium, then not including any whole-clusters is best. If using none of the stems, then don't use whole clusters.

Conclusions: This study was able to identify a small number of important factors and interactions that influence a Pinot Noir's flavor. Perhaps importantly, it has identified which factors do not have a significant influence. In line with the philosophy of sequential experimentation, this information can then be exploited in future studies.