

ASBC Annual Meeting

June 4–7 ■ Fort Myers, Florida

See what SCIENCE can brew for you

Simplify QAQC Analyses and Decision Making with Open Source Software

Drew Russey, Ph.D

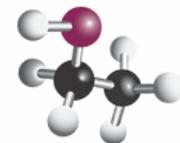
Laboratory Manager

drew_russey@saintarnold.com



Outline

- What is R?
- R in Brewery QAQC
- Case study: Brewhouse optimization
- Future Directions



The R Project for Statistical Computing

Pros

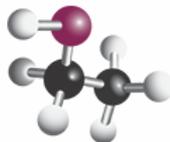
- Open source statistical software (FREE)
- Provides powerful, repeatable analyses
- Supplemental packages
- Efficient scripts are reusable and shareable tools
- Auxiliary open source software to improve user interface

Cons

- Steep learning curve
- Time investment



<https://www.r-project.org>



R Base Program



- Simple functions

```
mean() , sd() , sum()
```

- Complex operations e.g.

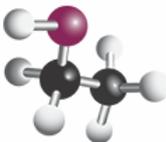
- Mean for all beer x

```
lapply(data.as.list, function(x) mean(x$interest))
```

- Match datasets by batch

- Merge data output from various equipment

```
data1[match(data2$batch, data1$batch)]
```



R Base Program



<https://www.r-project.org>

```

9 library(openxlsx)
10 library(stringr)
11 library(simpleboot)
12 library(gridExtra)
13 #### Custom Functions ####
14 #SG to Plato
15 sg2p<-function(sg) {
16   plato <- (-1 * 616.868) + (1111.14 * sg) - (630.272 * sg^2) + (135.997 * sg^3)
17 }
18
19 colNorm<-function(Data, COLUMN) {
20   list.data<-split(Data, Data$BRAND)
21   list.IBU<-lapply(list.data, function(x) (x[,COLUMN] - mean(x[,COLUMN], na.rm=TRUE)) / sd(x[,COLUMN], na.rm=TRUE))
22   data1<-do.call(rbind, list.data)
23   data2<-data.frame(data1$BATCH, as.vector(unlist(list.IBU)), stringsAsFactors = FALSE)
24   colnames(data2)<-c("BATCH", paste("Normalized",COLUMN, sep = "."))
25   arrange(data2, BATCH)
26 }
27
28 timeDiff<-function(data, COLUMN1, COLUMN2) {
29   temp.difference<-data[,COLUMN2] - data[,COLUMN1]
30   neg.times<-which(temp.difference<0)
31   tweak.times<-difftime((data[,COLUMN2] + 86400), data[,COLUMN1], units="secs")
32   sub.times<-replace(temp.difference, neg.times, tweak.times[neg.times])
33   return(as.numeric(sub.times)/60/60)
34 }
35
36 g_legend<-function(a.gplot){
37   tmp <- ggplot_gtable(ggplot_build(a.gplot))

```

mean {base}

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)
```

Default S3 method:

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

Arguments

x
An R object. Currently there are methods for numeric/logical v

R Console

```

> sg2p<-function(sg) {
+   plato <- (-1 * 616.868) + (1111.14 * sg) - (630.272 * sg^2) + (135.997 * sg^3)
+ }
>
> colNorm<-function(Data, COLUMN) {
+   list.data<-split(Data, Data$BRAND)
+   list.IBU<-lapply(list.data, function(x) (x[,COLUMN] - mean(x[,COLUMN], na.rm=TRUE)) / sd(x[,COLUMN],
+ na.rm=TRUE))
+   data1<-do.call(rbind, list.data)
+   data2<-data.frame(data1$BATCH, as.vector(unlist(list.IBU)), stringsAsFactors = FALSE)
+   colnames(data2)<-c("BATCH", paste("Normalized",COLUMN, sep = "."))
+   arrange(data2, BATCH)
+ }
>
> timeDiff<-function(data, COLUMN1, COLUMN2) {
+   temp.difference<-data[,COLUMN2] - data[,COLUMN1]
+   neg.times<-which(temp.difference<0)
+   tweak.times<-difftime((data[,COLUMN2] + 86400), data[,COLUMN1], units="secs")
+   sub.times<-replace(temp.difference, neg.times, tweak.times[neg.times])
+   return(as.numeric(sub.times)/60/60)
+ }
>
> g_legend<-function(a.gplot){
+   tmp <- ggplot_gtable(ggplot_build(a.gplot))
+   leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
+   legend <- tmp$grobs[[leg]]
+   return(legend)}
>
### Load in BrewData 2014
> CALCULATION0<-suppressWarnings(read_excel("/Users/brewerylab/Documents/R statistics/CSVs/BrewData2014.xlsx",
"CALCS", skip = 3))
> CALCULATION0$BRAND <- gsub(" ", "\\.", CALCULATION0$BRAND)
> CALCULATION0<-CALCULATION0[which(complete.cases(CALCULATION0$BATCH) ==TRUE),]
?mean
starting httpd help server ... done
> plot(runif(100,0,100), runif(100,0,100))

```

Useful Auxiliary Software



<https://www.r-project.org>

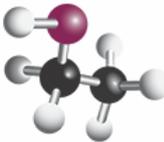
R: Base program

Rstudio: Aids workspace management

- one window instead of 2-4
- debugging support



<https://www.rstudio.com>



Useful Auxiliary Software



<https://www.rstudio.com>

RStudio

```

152 geom_text(x=1.5, y=4.9, label = "****", size=10)
153 p2<-ggplot(ELISSA3, aes(x=factor(ID), y=FIRST.RUNNINGS.BRIX)) + geom_boxplot(aes(fill = ID)) +
154 xlab(NULL) + ylab("First Runnings ( % O-P )") +
155 scale_fill_manual(values=c("white", "darkgray")) +
156 geom_text(x=1.5, y=24.5, label = "****", size=10)
157 p3<-ggplot(ELISSA3, aes(x=factor(ID), y=FINAL.RUNNINGS.BRIX)) + geom_boxplot(aes(fill = ID)) +
158 xlab(NULL) + ylab("Final Runnings ( % O-P )") +
159 scale_fill_manual(values=c("white", "darkgray")) +
160 geom_text(x=1.5, y=7, label = "****", size=10)
161 p4<-ggplot(ELISSA3, aes(x=factor(ID), y=TARGET.VOL.)) + geom_boxplot(aes(fill = ID)) +
162 xlab(NULL) + ylab("Target Volume (bbis)") +
163 scale_fill_manual(values=c("white", "darkgray"))
164
165 mylegend<-g_legend(p1)
166
167 case1<-grid.arrange(arrangeGrob(p4 + theme(legend.position="none"),
168 p1+ theme(legend.position="none"),
169 p2+ theme(legend.position="none"),
170 p3+ theme(legend.position="none"), nrow=2, ncol=2), mylegend, nrow=2, heights=c(10,1))
171
172
173
174
175
176 ~~~~~
177 ~~~~~
178
179 #add sorting columns
180 PUMP<-CALCULATION|CALCULATION$BRAND == "SABC.PUMPKINATOR",]
181 PUMP1<-PUMP[1:24,] %>%

```

Environment History

Global Environment		
confINT	function (Data)	
g_legend	function (a.gplot)	
mean_se2	function (x)	
mean_se3	function (x)	
sampLemean	function (x, d)	
SG.at.DH	function (calcol, d)	
sg2p	function (sg)	

Files Plots Packages Help Viewer

Zoom Export

Target Volume (bbis)

Control Treatment

First Runnings (% P)

Control Treatment

Console R Markdown

```

~/Documents/R statistics/R files/
> removed 2 rows containing non-finite values (stat_boxplot).
2: Removed 2 rows containing non-finite values (stat_boxplot).
3: Removed 4 rows containing non-finite values (stat_boxplot).
>
> case1
TableGrob (2 x 1) "arrange": 2 grobs
z cells name grob
1 1 (1-1,1-1) arrange gtable[arrange]
2 2 (2-2,1-1) arrange gtable[guide-box]
Warning message:
In eval(formal.args[[as.character(substitute(arg))]]) :
reached elapsed time limit
> grid.arrange(arrangeGrob(p4 + theme(legend.position="none"),
+ p1+ theme(legend.position="none"),
+ p2+ theme(legend.position="none"),
+ p3+ theme(legend.position="none"), nrow=2, ncol=2), mylegend, nrow=2, heights=c(10,1))
Warning messages:
1: Removed 1 rows containing non-finite values (stat_boxplot).
2: Removed 2 rows containing non-finite values (stat_boxplot).
3: Removed 4 rows containing non-finite values (stat_boxplot).
>

```

Useful Auxiliary Software



<https://www.r-project.org>

R: Base program

Rstudio: Aids workspace management

- one window instead of 2-4
- debugging support



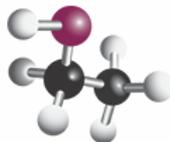
<https://www.rstudio.com>

Rshiny: build web applications using only R code. When finished...

- Can perform analyses with pull down menus
- Others can use without ever seeing code.



<http://shiny.rstudio.com>



Useful Auxiliary Software

~/Documents/R statistics/Dashboard QAQC - Shiny

http://127.0.0.1:3081 Open in Browser Publish

Saint Arnold Brewing Company #SAF

- Single Batch Data
- Current Fermentations
- Trend Checking and Analysis
- Regression
- Histogram
- t-tests (two groups)
- Barrels
- Default Settings

Inputs

Select a yeast

Beer Brand

Select Variable

Select a batch

- 15-032
- 15-037
- 15-080
- 15-081
- 15-116
- 15-117
- 15-130
- 15-131

Boxplot Comparison

is.selected
■ not selected
■ selected

IBU

Not Selected vs Selected?

Data Summary

	Var.2	Selected	Not Selected
1	Max	67.1	86.9
2	95% upr	68.2164940165529	77.5246553623457
3	Mean	61.73	75.34
4	95% lwr	55.2501726501137	73.1642335265432
5	Min	55.7	63.2

Showing 1 to 5 of 5 entries

-13.6

Difference of Means

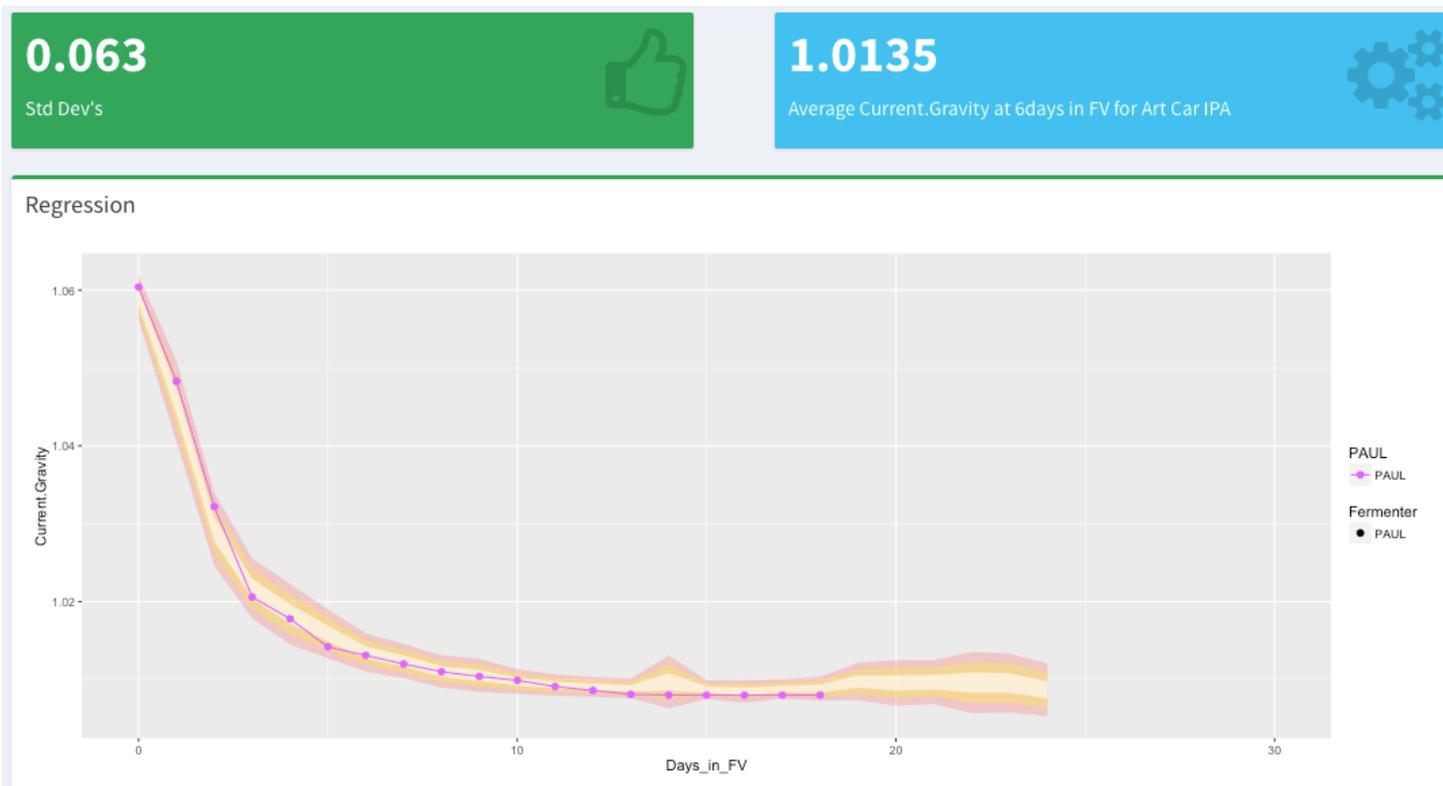
0.0385

p value



R Use Examples

- Track active fermentations
- Check short term and long term trends
- Generate monthly report pdf's
- **Analyze experiments**



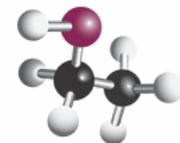
Data Points

	Product	Batch	Fermenter	Days_in_FV	Current.Gravity
	Art Car IPA	17099	PAUL	6	1.0131

Showing 1 to 1 of 1 entries

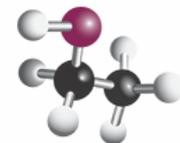
Experimental Design

- Step 1: Observation
- Step 2: Form Hypothesis
- Step 3: Plan experiment
 - Ensure results are quantifiable
- Step 4: Power Analysis
 - Aids planning process changes
 - Know when to expect results (determine n)
- Step 5: Execute n trials
- Step 6: Statistically analyze results of n trials
 - p -value will provide unbiased answer (significance)
 - d will indicate the magnitude of change



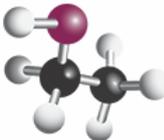
R in the Lab

- Step 1: Observation
- Step 2: Form Hypothesis
- Step 3: Plan experiment
 - Ensure results are quantifiable
- **Step 4: Power Analysis**
 - Aids planning process changes
 - Know when to expect results (determine n)
- Step 5: Execute n trials
- **Step 6: Statistically analyze results of n trials**
 - p -value will provide unbiased answer
 - d will indicate the magnitude of change



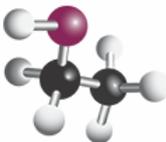
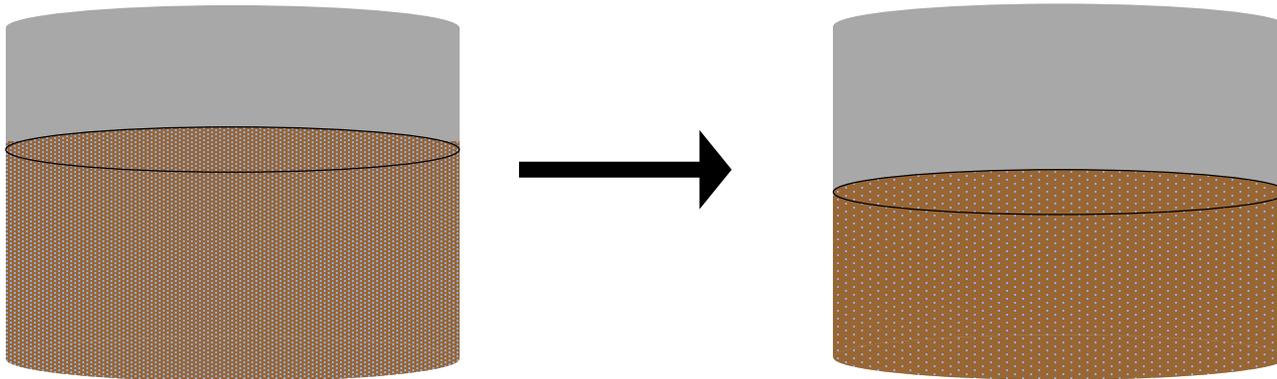
Brewhouse Optimization: Observation

- **Step 1: Observation**
 - Lauter tun designed to make low Plato beers. Inefficient with higher gravities (IPA).
 - Final runnings were around 5°P



Brewhouse Optimization: Hypothesis

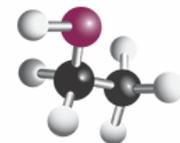
- Step 2: **Hypothesis** Use less grain and thicken mash by reducing mash in water
- Adjusting allocation of water from mash to sparge to more efficiently extract



Brewhouse Optimization: Plan Method

- **Step 3: Plan experiment**
 - Liquor to grist ratio (liquor reduced 6%)
 - Reduce grain bill ~5% to accommodate higher efficiency
 - Drain 14% more wort prior to sparge
 - Increase sparge liquor volume (6%)
 - Calculate Brewhouse Efficiency (Wort Extract/Potential Extract)

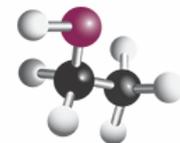
For full details: Gutierrez and Russey, 2017, Improving Brewhouse Efficiency by Adjusting Mash Thickness and Lauter and Sparge Volumes



Brewhouse Optimization with R

- **Step 4: Power Analysis**
 - **Power = probability of finding a real effect**
 - `pwr` package in R will solve for n_2 `pwr.t2n.test()`
 - Requires 4 of the 5 below, will solve for remaining variable
 - n_1 = Control group
 - n_2 = Treatment group
 - d = effect size
 - P = power. Typically 0.8
 - α = Significance level. Typically 0.05

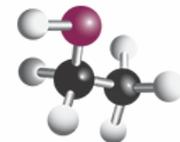
```
pwr.t2n.test(n1=40, p=0.8, sig.level=0.05, ...)
```



Brewhouse Optimization with R

- **Step 4: Power Analysis**
 - **Power = probability of finding a real effect**
 - `pwr` package in R will solve for n_2 `pwr.t2n.test()`
 - Requires 4 of the 5 below, will solve for remaining variable
 - n_1 = Control group
 - n_2 = Treatment group
 - d = effect size
 - P = power. Typically 0.8
 - α = Significance level. Typically 0.05

```
pwr.t2n.test(n1=40, p=0.8, sig.level=0.05, ...)
```

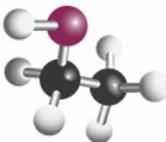


Effect Size (d)

- Many ways to calculate, but essentially...
the shift in means as standard deviations

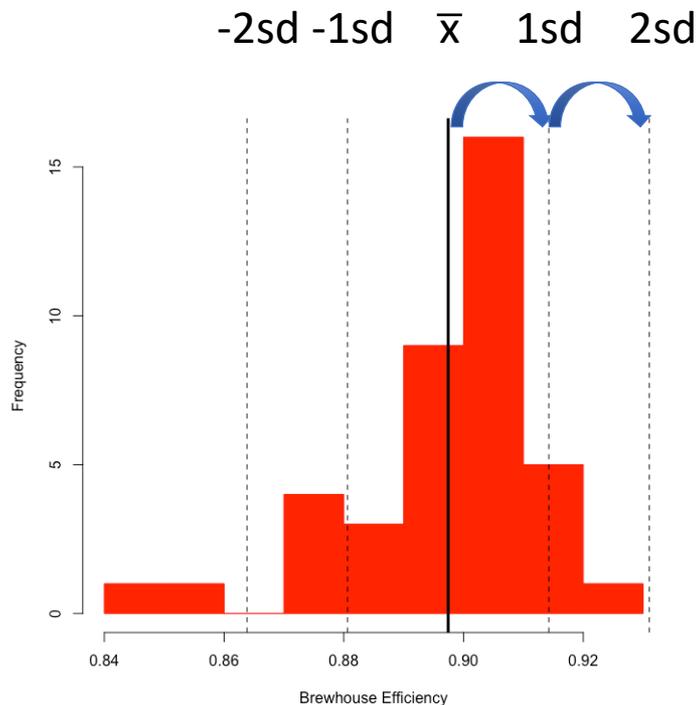
$$d = \frac{\mu_{\text{Treatment}} - \mu_{\text{control}}}{\sigma_{\text{Control}}}$$

```
(mean(treatment) - mean(control)) / sd(control)
```



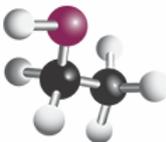
Effect Size (d)

- Many ways to calculate, but essentially...
the shift in means as standard deviations

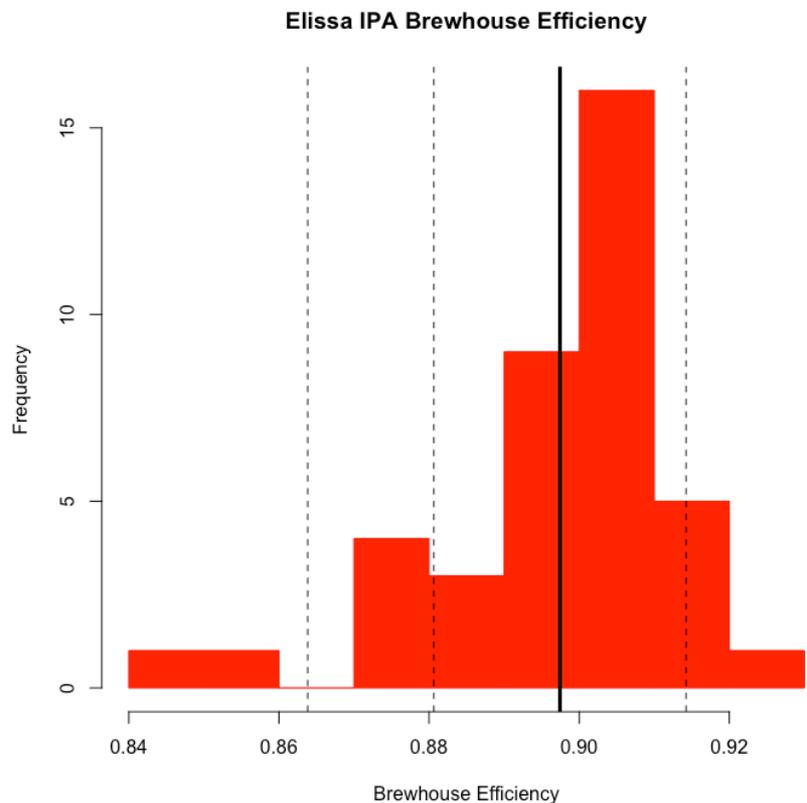


$$d = \frac{\mu_{\text{Treatment}} - \mu_{\text{Control}}}{\sigma_{\text{Control}}}$$

A.K.A. How many standard deviations will new treatment batches be from controls



Brewhouse Optimization with R



• Step 4: Power Analysis

Determine when to expect results
(determine needed n)

- Variables for `pwr.t2n.test()`

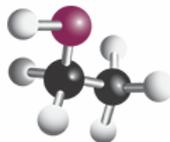
$n_1 = 40$ mashes from previous year

$P = 0.8$

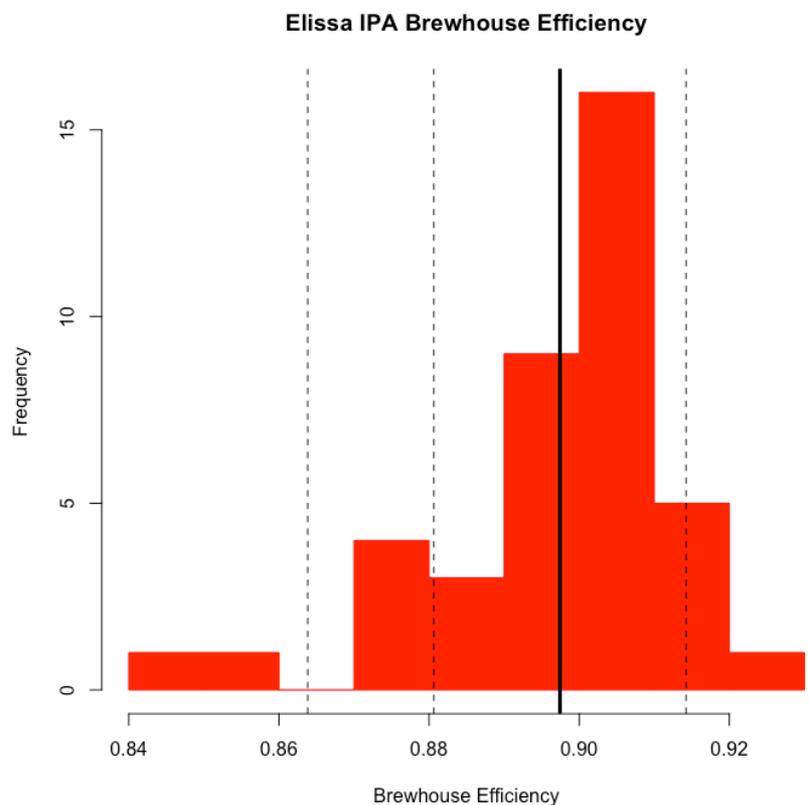
$\alpha = 0.05$

$d =$

Description	d	n_2 needed
Small	0.3	∞
Medium	0.5	153
Large	0.8	19
Huge	2.0	3



Brewhouse Optimization with R

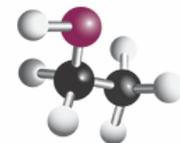


• Step 4: Power Analysis

Determine how large of an effect you could find

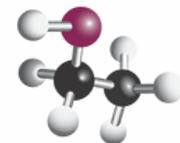
- Variables for `pwr.t2n.test()`
 - $n_1 = 40$ mashes from previous year
 - $P = 0.8$
 - $\alpha = 0.05$
 - $n_2 =$

n_2	Min d	Description
3	1.71	Large
6	1.25	Large
10	1.01	Large
34*	0.66	Medium



Brewhouse Optimization with R

- Step 1: Lauter tun designed to make low Plato beers. Inefficient with higher gravities such as IPA.
- Step 2: Adjusting allocation of water from mash to sparge will enhance brewhouse efficiency
- Step 3: Reallocate water from mash to sparge, condense grist, quantify brewhouse efficiency
- Step 4: Power Analysis
 - Determine number of trials necessary and/or
 - Determine how large an effect needed given n trials
- **Step 5: Execute**
- Step 6: Analyze results with t-test equivalent (Mann-Whitney u-test)



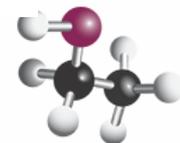
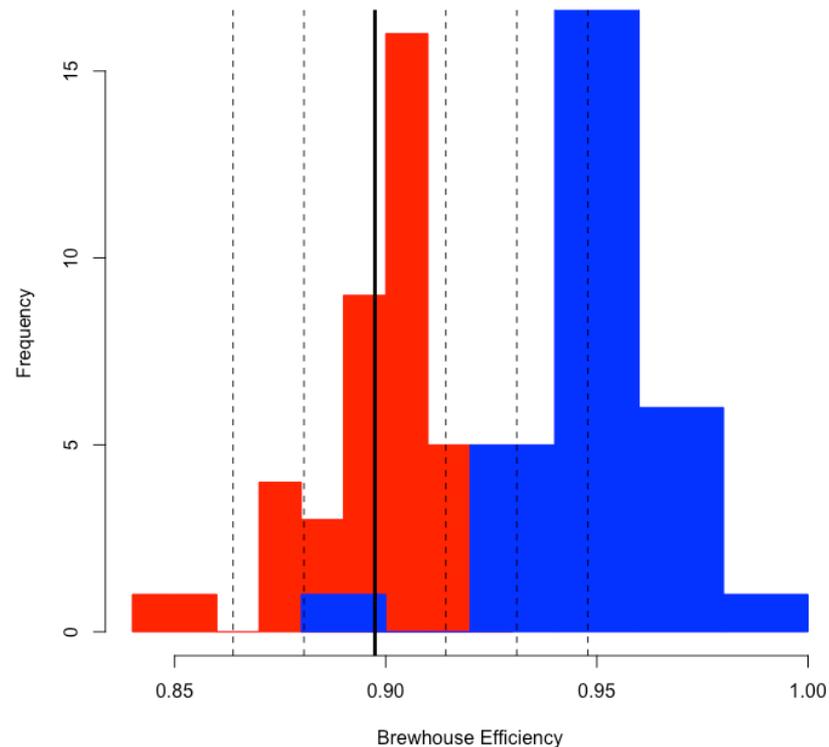
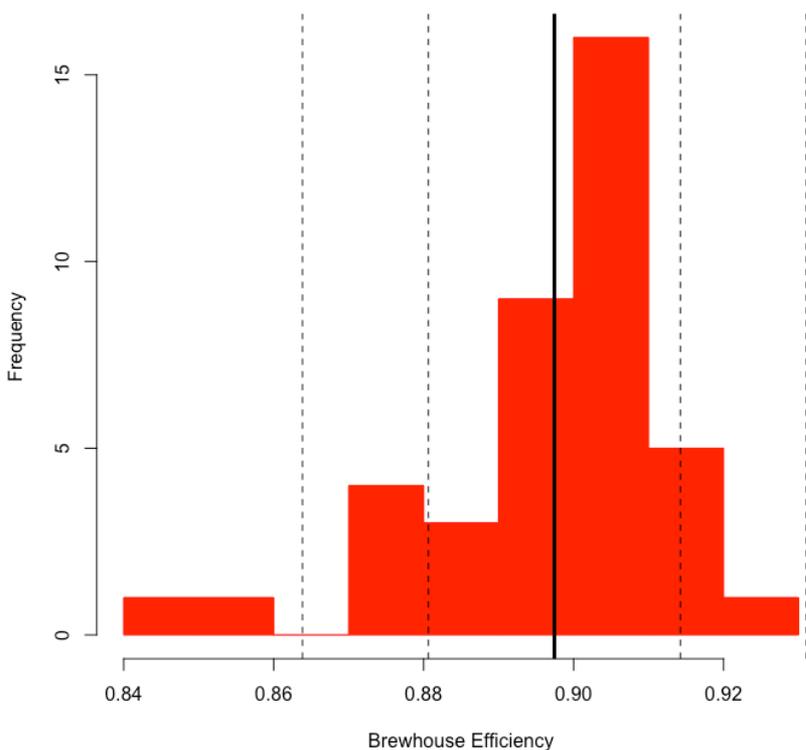
Increase in Brewhouse Efficiency

-2sd -1sd x 1sd 2sd

89.7% 94.9%

Elissa IPA Brewhouse Efficiency

Elissa IPA Brewhouse Efficiency

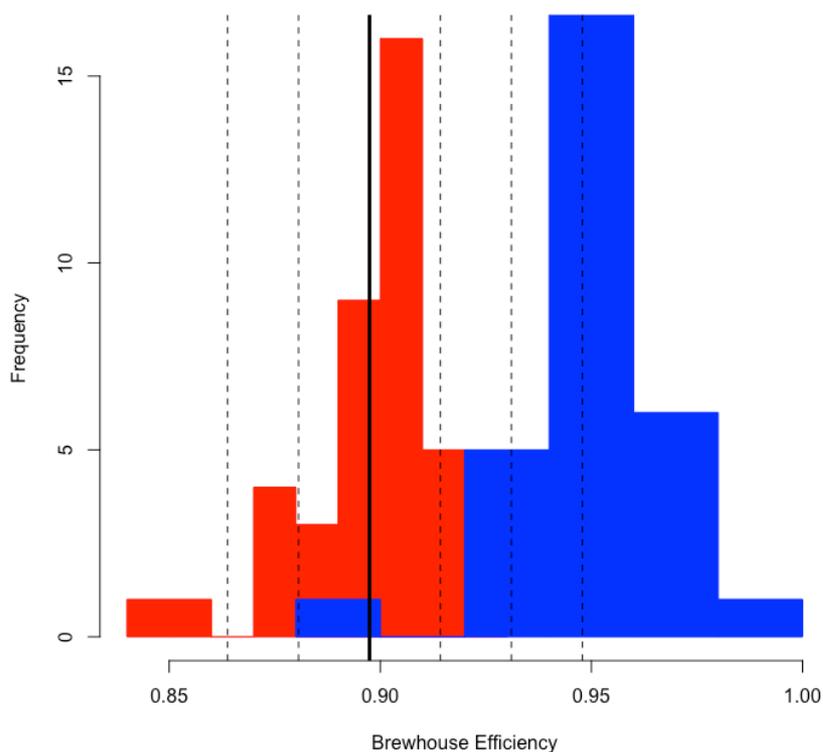


Increase in Brewhouse Efficiency

$$1 + 1 + 1 = d = 3$$



Elissa IPA Brewhouse Efficiency

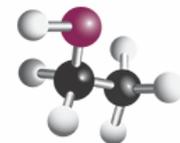


$$d = \frac{\mu_{\text{Treatment}} - \mu_{\text{Control}}}{\sigma_{\text{Control}}}$$

$$d = \frac{94.9\% - 89.7\%}{1.68\%}$$

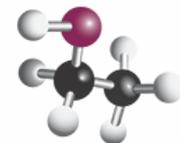
$d = 3.09$ a “huge” effect

But is it significant?

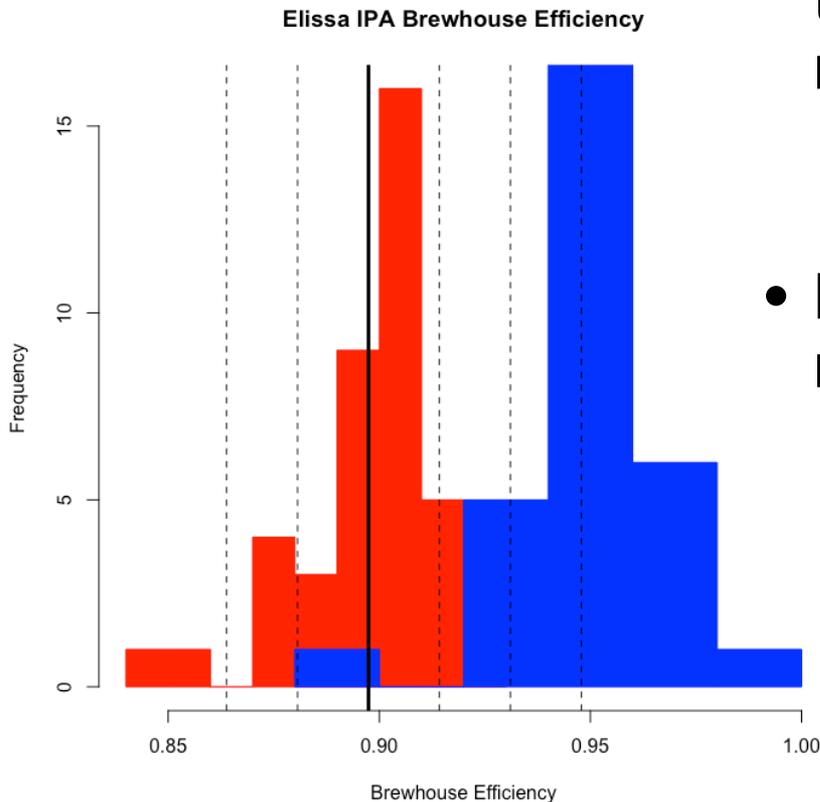


Brewhouse Optimization with R

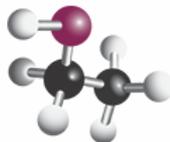
- Step 1: Lauter tun designed to make low Plato beers. Inefficient with higher gravities such as IPA.
- Step 2: Adjusting allocation of water from mash to sparge will enhance brewhouse efficiency
- Step 3: Reallocate water from mash to sparge, condense grist, quantify brewhouse efficiency
- Step 4: Power Analysis
 - Determine number of trials necessary and/or
 - Determine how large an effect needed given n trials
- Step 5: Execute
- Step 6: Analyze results with t-test equivalent (Mann-Whitney u-test)



Increase in Brewhouse Efficiency



- Shapiro test on both control and treatment groups to test for normality
 - `shapiro.test(efficiency)`
 - $p = < 0.05$, non-normal
- Mann-Whitney/Wilcoxon test is a non-parametric t-test equivalent
 - Safe to use with non-normal distribution
 - `wilcox.test(efficiencyred, efficiencyblue)`
 - $p = 2.45 \times 10^{-12}$

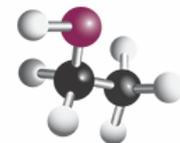




FutuRe DiRections

<https://www.r-project.org>

- Increase awareness of open source software for brewing and brewery lab analyses
- Create a community of R users to share targeted scripts
 - Make analyses accessible to breweries of all sizes
 - Standardize methods of analysis alongside measurement
 - Increase reproducibility and transparency of beer science
- All base R scripts from this presentation available



Questions?



drew_russey@saintarnold.com

Special Thanks to

Eddie Gutierrez
Saint Arnold Brewing Company
eddie_gutierrez@saintarnold.com

2017 ASBC Meeting

