Modeling The Freshening Power of the Hop

Version 0.21 1/27/19

merge from ASBC2019_lrcPart1_model1 tidy transform_ujbc_a_1469081_sm5496.txt.v05_add exptOLD_thisWORKS.RMD and ASBC2019_lrcPart2_model1_ujbc_a_1469081_sm5496.txt.v12b.RMD

ActionItem: reforat from scratch following https://labrtorian.com/tag/cran/ https://labrtorian.com/tag/cran/

Step1: Open this .RMD file in R Studio and click the "Preview" button above. A formatted version should pop up in a browser window. IF not, troubleshoot that!

outline

Intro

This is primarily an excercise in data wrangling, modeling, and markdown formatting using R Studio. In that context, this analysis is a *preliminary* attempt to develop a metric for *Hop Freshening Power* (syn. 'dryhop creep', 'ABV creep', 'dry-hop creep') and to model this aspect of the process of making dry-hopped beers. If you care what this means in a beer and brewing context, bear in mind that several research groups the Shellhammer group at Oregon State Universityhas done extensive work .

In contrast to much of the recent work to characterize the molecular mechanisms at play, the data used for this work are relatively 'simple' whole-system factors and endpoints: a certain amount of hops are added to beers under various conditions, allowed to react for various amounts of time, then the resulting beers were tested for alcohol content (using Anton Paar DMA4500 Beer Alcolyzer) to produce the data we are using in this exercise.

load libraries:

Have a look at the data:

```
mydata <-read.csv("ujbc_a_1469081_sm5496.txt")  ## SPECIFY filename
# various ways to select columns by data type
#mydata %>% select_if(negate(is.numeric)) %>% names() ## NOT nonnumericcols (NOTE: integers!)
#mydata %>% select_if(funs(is.integer(.))) %>% names() ## integers (note funs() wrapper)
#mydata %>% select_if(~ !(is.integer(.x)) | is.numeric(.x)) %>% names() ## num OR int
#mydata %>% select_if(~ !(is.integer(.)) | is.numeric(.)) %>% names() ## num OR int
#mydata %>% select_if((is.factor)) %>% names() ## factors
#mydata %>% select_if(is.numeric) %>% names() ## numeric (NOTE: integers included!)
#mydata %>% select_if(is.character()) %>% names() ## characters
names(mydata)
```

```
##  [1] "sample_id"              "hop"
##  [3] "BINhop"                 "expt"
##  [5] "sample_group"           "special_group"
##  [7] "brew_date"              "sample_collection_date"
##  [9] "volume_mL"              "dryhop_date"
## [11] "dhop_day"               "Hop_type"
## [13] "mg_hops"                "contact_days"
## [15] "special_conditions"     "temp.C"
```

```
## [17] "Test_Date"              "Test.time"
## [19] "ABV"                    "ABW"
## [21] "OE"                     "Er"
## [23] "Ea"                     "SG"
## [25] "RDF"                    "ADF"
## [27] "Calories"               "REF_NH"
## [29] "ABV_increase"           "vol_norm_ABV_increase"
## [31] "ABV_increase_prototabs"
```

```
table(mydata$Hop_type)
```

```
##
##        AMAR        CASC      CENT14    CENT14OX      CENT15    CENT15OX
##           6           6           3           3           3           3
##      CENT16 CENT16Grind    CENT16OX  CENT17Cone         CIT          NH
##          72           3           3           3           6          60
##         SIM
##           6
```

Several different columns in the original dataset were evaluated as specific endpoints (Y-values) to model the *freshening power of the hop*. Through that, two errors in the original approach led to overly-complicated (and, of course, mostly worthless :) models. a focus on ABV rather than ABW, and the notion that the metric was a property of the hops, rather than a property of the entire system. Metrics equating the increase in ethanol (FPH_EtOH) or carbon dioxide (FPH_CO2) produced to the amount of hops added (all in g/100mL), relative to the same system with no dry-hopping. The metric was developed through attempts to correct for the exact amount of hops added (w/v) in each experiment.

Regardless how we measure this phenomenon, through experience in brewing we know that:

Endpoints of dryhopping * flavor impact (always) * impact on visual/presentation (sometimes) * *ethanol increase* accompanied by decrease in specific gravity (sometimes) * CO2 increase (sometimes) * diacetyl/VDK increase (sometimes) * and so on…

Focusing on *ethanol increase as the endpoint*, we know these to be *relevant factors* (and comments as they relate to these data): * variety of hops (five different varieties) * presence of live yeast (true in all cases for these data) * temp.C = temperature during dryhopping (mostly warm, with a few in the cold) * amount of contact time between hops and beer (up to 7 weeks)

We will ultimately widdle down the input data to only include Centennial hops and evaluate several functions that equate ethanol and **calculated CO2** increases separately as functions of contact time.

table of useful models with examples...

It should be *noted that CO2 is somewhat toxic to yeast, but this probably had minimal effect on these data because the experiments were carried out in vented (crimped foil) containers. Had these reactions carried out in closed containers, it is unlikely they would have proceeded in this manner. Some of the calculated CO2 concentrations below would presumably impact almost any yeast-catalyzed reaction.

In most cases the variables used below to model hop freshening power are *calculations* based on the original data. The process of creating new variables from existing variables is known as *data transformation*, one of Wickham's four iterations of data analysis. The process of transforming the original data was detailed in a preceeding Notebook and presented below as a single code chunk.

For more details on this dataset see the manuscript. Undoubtedly, any 'complete' model of *Hop Freshening Power* will incorporate molecular information exemplified by the OSU researchers (e.g. production of glucose and maltose) and will certainly involve some yeast-related factors.

- please send complaints and corrections to luke.chadwick@gmail.com!
- Data are from the file "ujbc_a_1469081_sm5496.txt" (supplementary data from Jacob A. Kirkendall, Carter A. Mitchell & Lucas R. Chadwick (2018): The Freshening Power of Centennial Hops, *Journal of the American Society of Brewing Chemists* Volume 76, Issue 3, Pages 178-184 (**2018**) DOI: 10.1080/03610470.2018.1469081* available from https://www.tandfonline.com/doi/full/10.1080/03610470.2018.1469081?scroll=top&needAccess=true
- Wickham's four pillars of data analysis:
  - tidy
    - transform
    - visualize
    - *model*
- *"Data is not information. Information is not knowledge. And knowledge is certainly not wisdom."*
- [this is not a] stats class
  - descriptive statistics
  - inferential statistics
  - *modeling –> the journey as the destination*

FPHcalc in a single chunk: chunk_tidytransform

data summary

```
## # A tibble: 19 x 7
##    EXPTnew        hops_g_100mL pounds_bbl ABW_increase FPH_EtOH FPH_CO2
##    <fct>                 <dbl>      <dbl>        <dbl>    <dbl>   <dbl>
##  1 group2A42TRUE         0.374      0.965       0.78      2.08    1.99
##  2 group2A33TRUE         0.386      0.994       0.743     1.93    1.85
##  3 group2A42FALSE        0.398      1.03        0.763     1.92    1.84
##  4 group2A33FALSE        0.393      1.01        0.713     1.81    1.73
##  5 group2A25TRUE         0.380      0.978       0.673     1.78    1.70
##  6 group2A25FALSE        0.393      1.01        0.68      1.73    1.66
##  7 group2A19TRUE         0.367      0.945       0.59      1.61    1.54
##  8 group2A19FALSE        0.390      1.00        0.597     1.53    1.46
##  9 group2A12FALSE        0.402      1.04        0.537     1.34    1.28
## 10 group2A12TRUE         0.362      0.934       0.457     1.26    1.21
## 11 group4BFALSE          0.365      0.940       0.384     1.05    1.01
## 12 group3AFALSE          0.408      1.05        0.333     0.814   0.779
## 13 group1AFALSE          0.387      0.997       0.286     0.739   0.707
## 14 group1BFALSE          0.385      0.993       0.279     0.732   0.700
## 15 group2A05TRUE         0.374      0.964       0.223     0.598   0.572
## 16 group2A05FALSE        0.382      0.984       0.223     0.585   0.560
## 17 group2A01FALSE        0.398      1.02        0.0533    0.134   0.128
## 18 group2A01TRUE         0.370      0.954       0.0467    0.127   0.121
## 19 group3BFALSE          0.418      1.08        0.05      0.115   0.110
## # ... with 1 more variable: calcCO2vols_increase <dbl>
```

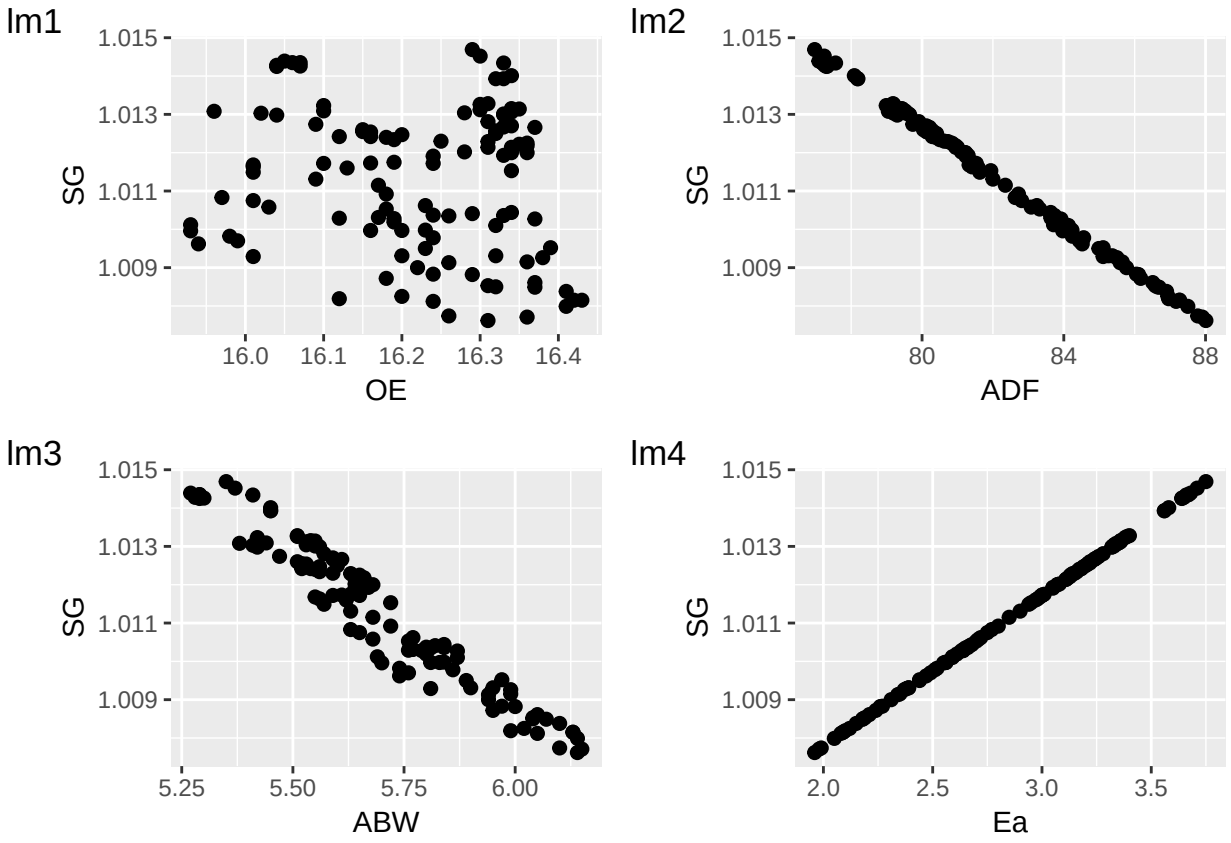
modeling Overview

Observing the impacts of dryhopping in the presence of live yeast has led many brewing professionals to understand that FPH is a function of many of the variables above including hop variety,form_of_hops,harvestYear,OX,DH_temp,daysonhops,rouse,pounds_bbl…. Many have intuitively created a model in their heads (without necessarily thinking of it as such) and skillfully adjust process when necessary to account for this phenomenon. In linear modeling, our function will take on the form: $FPH = intercept + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$ where β values are what we're attempting to derive in this modeling exercise, and X values are (collectively) a particular set of conditions.

ActionItem: add link to modeling overview

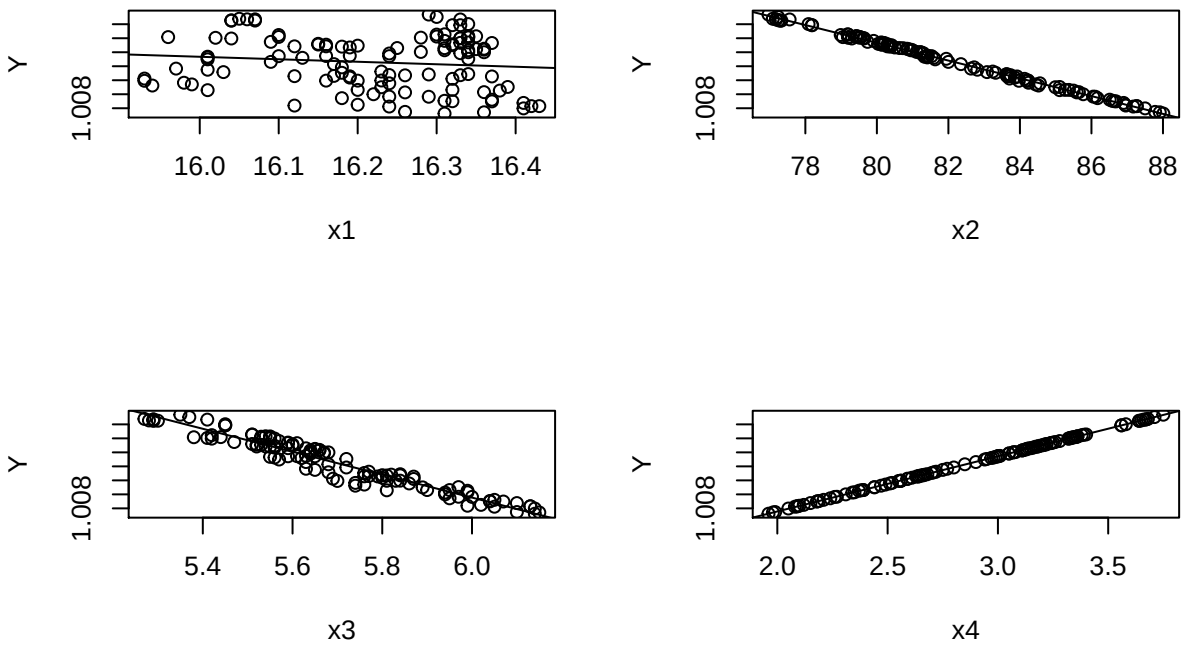
vector~vector scatter plots

```
FPHcalc <- read.csv("FPHcalc.csv", stringsAsFactors = TRUE)

df<- FPHcalc
p1<- ggplot(df, aes(y=SG,x=OE)) + geom_point(size=2) + labs(tag="lm1")
p2<- ggplot(df, aes(y=SG,x=ADF)) + geom_point(size=2) + labs(tag="lm2")
p3<- ggplot(df, aes(y=SG,x=ABW)) + geom_point(size=2) + labs(tag="lm3")
p4<- ggplot(df, aes(y=SG,x=Ea)) + geom_point(size=2) + labs(tag="lm4")
grid.arrange(p1, p2, p3, p4, ncol = 2)
```



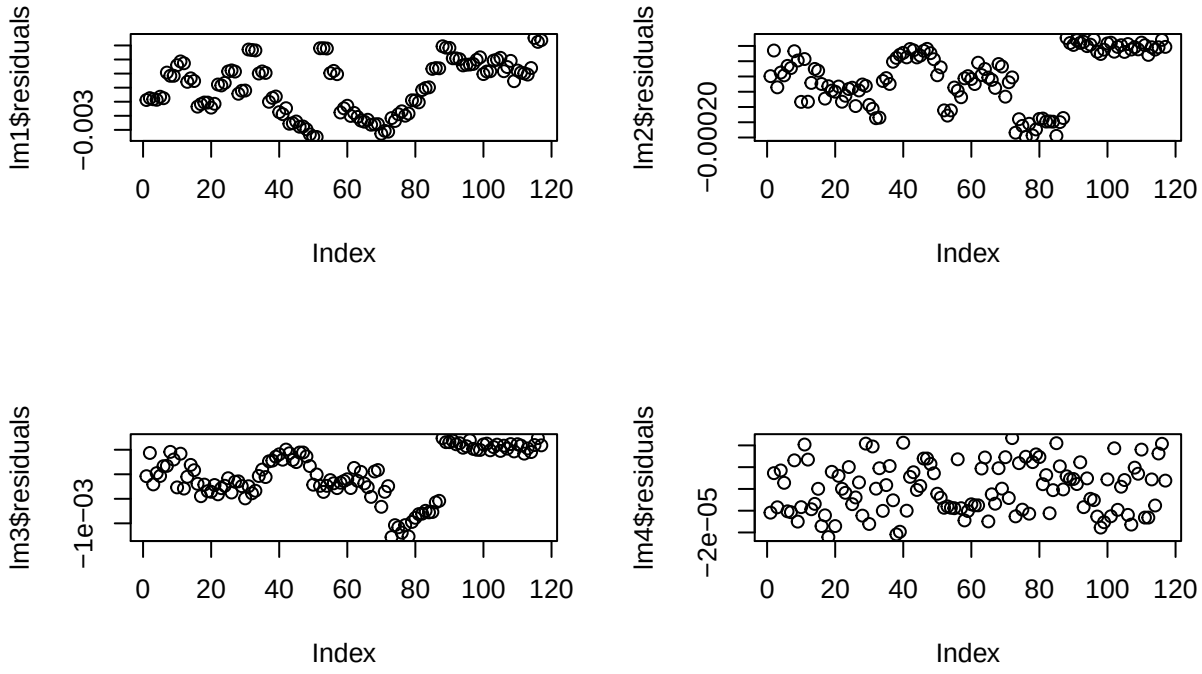
linear models 1 (create models)

```
df<-FPHcalc
Y <- df$SG
x1 <- df$OE
x2 <- df$ADF
x3 <- df$ABW
x4 <- df$Ea

lm1<-lm(Y~x1)
lm2<-lm(Y~x2)
lm3<-lm(Y~x3)
lm4<-lm(Y~x4)
par(mfrow = c(2, 2), oma = c(0, 0, 2, 0))
plot(Y~x1)
abline(lm1)
plot(Y~x2)
abline(lm2)
plot(Y~x3)
abline(lm3)
plot(Y~x4)
abline(lm4)
```



linear models 2 (view residual plots)

```
par(mfrow = c(2, 2), oma = c(0, 0, 2, 0))
plot(lm1$residuals)
plot(lm2$residuals)
plot(lm3$residuals)
plot(lm4$residuals)
```




```
#summary(lm4)
```

compare linear models with mtable function in package memisc

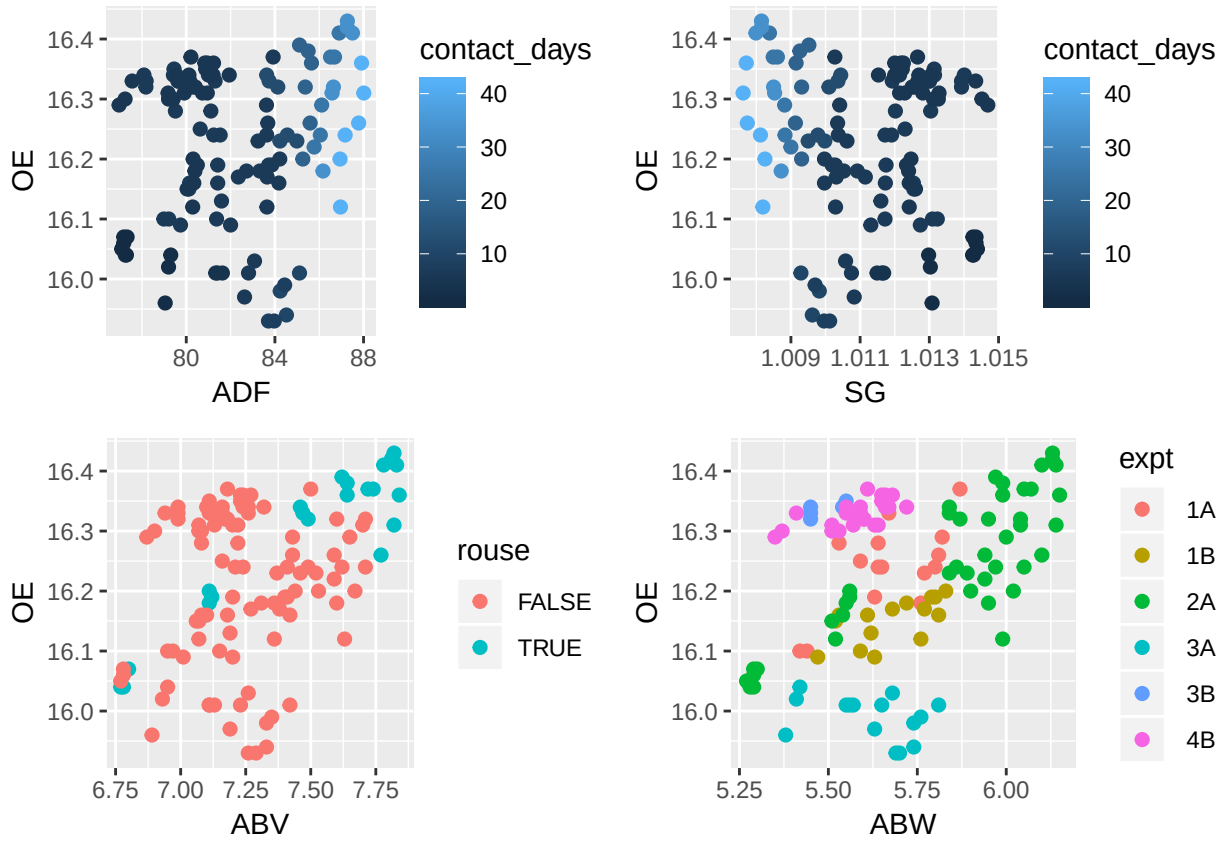
```
mtable1234 <- mtable("Model 1"=lm1,"Model 2"=lm2,"Model 3"=lm3, "Model 4"=lm4,
                     summary.stats=c("sigma","R-squared","F","p","N"),show.eqnames=T)

mtable1234b <- relabel(mtable1234,
                       "(Intercept)" = "Constant",
                       OE = "OE = Original Extract (g/100mL)",
                       ADF = "ADF = Apparent Degree of Fermentation (%)",
                       ABW = "ABW = Ethanol (w/w)",
                       Er = "Er = Residual Extract (g/100mL)"
                       )
mtable1234
```

```
##
## Calls:
## Model 1: lm(formula = Y ~ x1)
## Model 2: lm(formula = Y ~ x2)
## Model 3: lm(formula = Y ~ x3)
## Model 4: lm(formula = Y ~ x4)
##
## ============================================================================
##                  Model 1      Model 2       Model 3       Model 4
##                ----------- ------------- ------------ ---------------
##                     Y            Y             Y             Y
## ----------------------------------------------------------------------------
##   (Intercept)    1.040***      1.063***     1.058***        1.000***
##                 (0.022)       (0.000)      (0.001)         (0.000)
##   x1            -0.002
##                 (0.001)
##   x2                          -0.001***
##                               (0.000)
##   x3                                       -0.008***
##                                            (0.000)
##   x4                                                        0.004***
##                                                            (0.000)
## ----------------------------------------------------------------------------
##   sigma           0.002         0.000        0.001           0.000
##   R-squared       0.015         0.998        0.929           1.000
##   F               1.734     49328.418     1502.585     3152214.958
##   p               0.191         0.000        0.000           0.000
##   N             117           117          117             117
## ============================================================================
```

```
#show_html(mtable1234b)
```

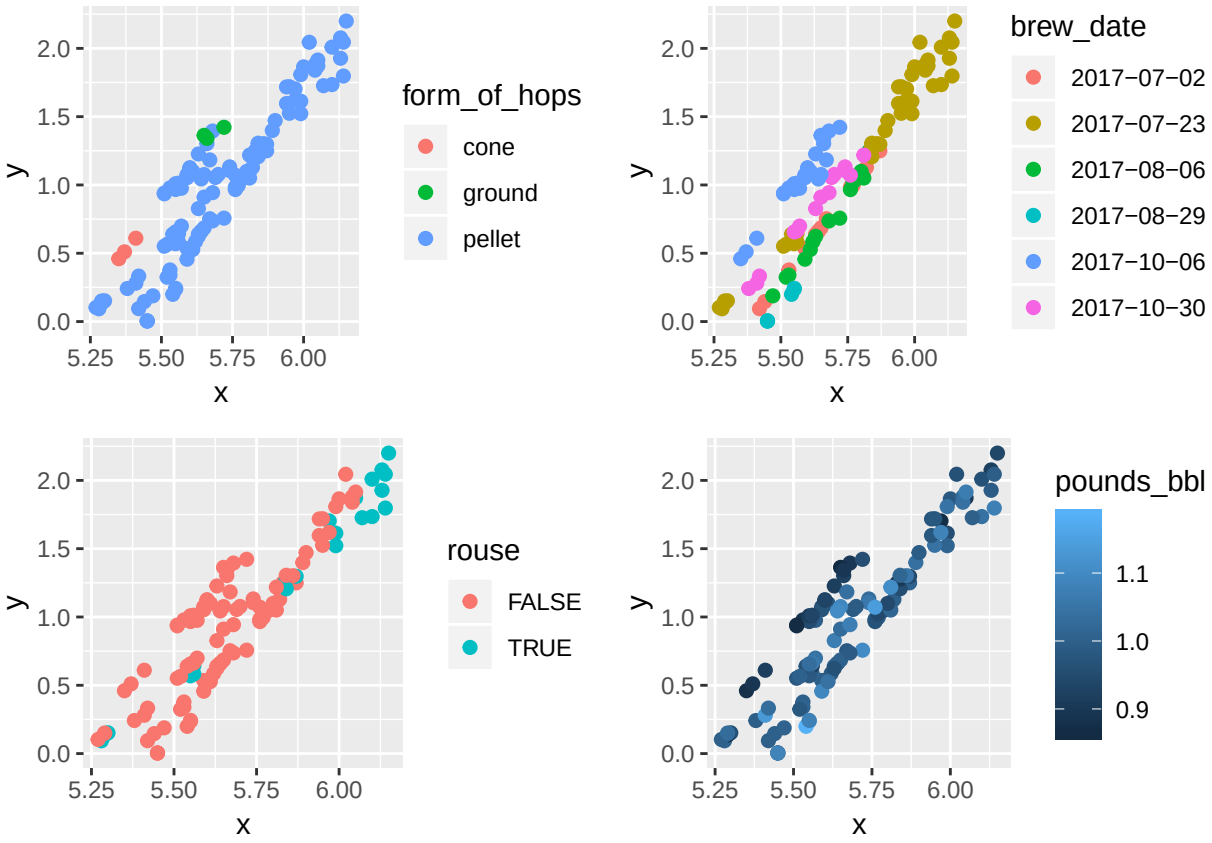
vector~vector*vector plots

```
df <-FPHcalc
p1<- ggplot(df, aes(y=OE,x=ADF, color=contact_days)) + geom_point(size=2)
p2<- ggplot(df, aes(y=OE,x=SG, color=contact_days)) + geom_point(size=2)
p3<- ggplot(df, aes(y=OE,x=ABV, color=rouse)) + geom_point(size=2)
p4<- ggplot(df, aes(y=OE,x=ABW, color=expt)) + geom_point(size=2)
grid.arrange(p1, p2, p3, p4, ncol = 2)
```



x~y*(4 vectors) by color

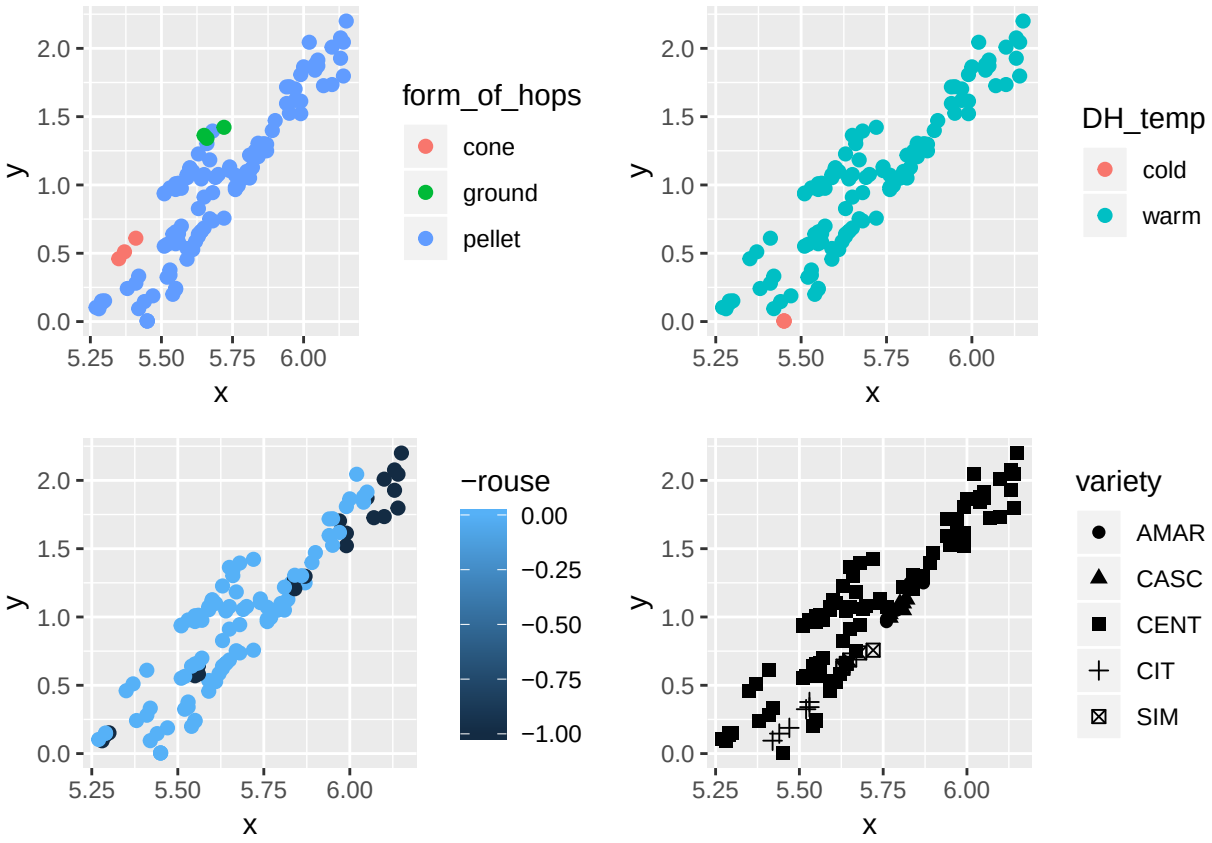
```
df <- FPHcalc
x<- df$ABW
y<- df$FPH_EtOH

p1<- ggplot(df, aes(x,y, color=form_of_hops)) + geom_point(size=2)
p2<- ggplot(df, aes(x,y, color=brew_date)) + geom_point(size=2)
p3<- ggplot(df, aes(x,y, color=rouse)) + geom_point(size=2)
p4<- ggplot(df, aes(x,y, color=pounds_bbl)) + geom_point(size=2)
grid.arrange(p1, p2, p3, p4, ncol = 2)
```



!! include brew_date in model

x~y*vector by shape

```
df <- FPHcalc
x<- df$ABW
y<- df$FPH_EtOH

p1<- ggplot(df, aes(x,y, color=form_of_hops)) + geom_point(size=2)
p2<- ggplot(df, aes(x,y, color=DH_temp)) + geom_point(size=2)
p3<- ggplot(df, aes(x,y, color=-rouse)) + geom_point(size=2)
p4<- ggplot(df, aes(x,y, shape=variety)) + geom_point(size=2)
grid.arrange(p1, p2, p3, p4, ncol = 2)
```



Amunategui "Using Correlations To Understand Your Data"

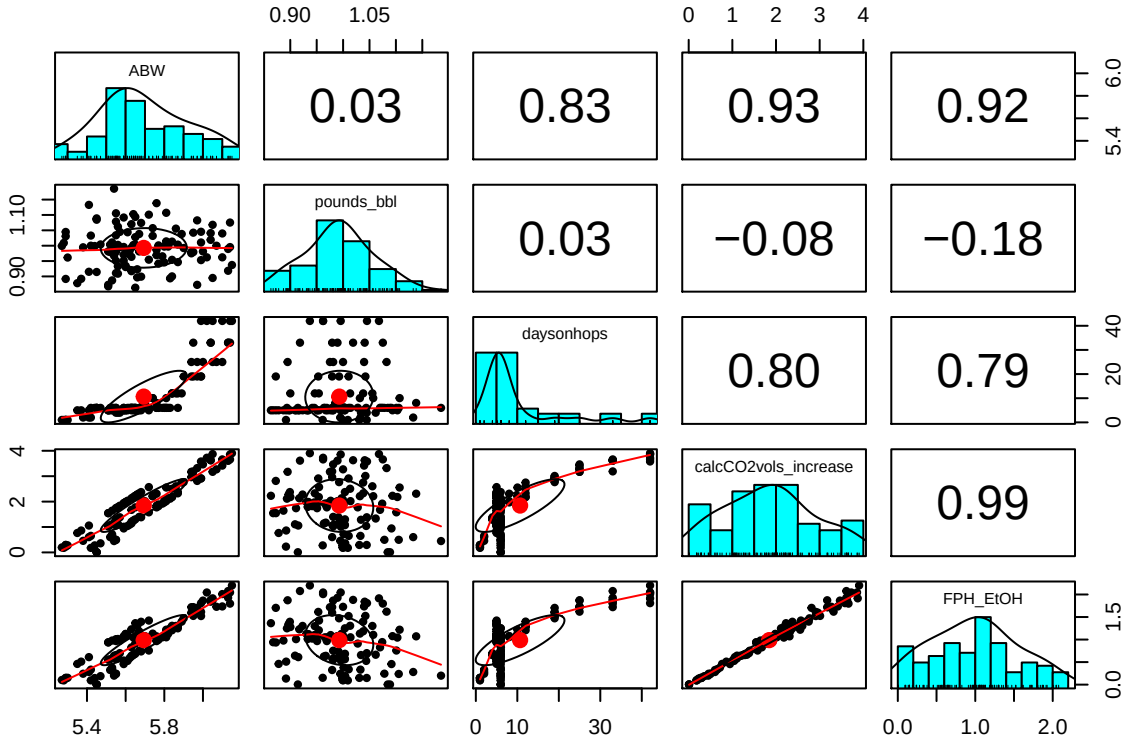
```
mydata<-FPHcalc %>% dplyr::select(variety,form_of_hops,pounds_bbl,rouse,daysonhops,DH_temp,ABW,SG,OE,AD

## note use of "dplyr::select" because one of these packages is conflicting with dplyr commands :()

## following  Manuel Amunategui  https://www.youtube.com/watch?v=igPQ-pI8Bjo
## Using Correlations To Understand Your Data: Machine Learning With R
##functions for flattenSquareMatrix
cor.prob <- function (X, dfr=nrow(X) -2) {
  R<- cor(X, use="pairwise.complete.obs")
  above<- row(R) < col(R)
  r2 <- R[above]^2
  Fstat<- r2 * dfr/(1-r2)
  R[above] <- 1- pf(Fstat, 1, dfr)
  R[row(R) == col(R)] <- NA
  R
}
flattenSquareMatrix <- function(m) {
  if( (class(m) != "matrix") | (nrow(m)!=ncol(m))) stop("Must be a square matrix.")
  if(!identical(rownames(m), colnames(m))) stop("Row and column names must be equal.")
  ut <- upper.tri(m)
  data.frame(i = rownames(m)[row(m)[ut]],
             j = rownames(m)[col(m)[ut]],
```

```
            cor=t(m)[ut],
            p=m[ut])
}

## library(caret) to dummify everything (turn all characters&factors into columns;  ignores numbers and
dmy<- dummyVars(" ~ .",data = mydata)
mydummifieddata<- data.frame(predict(dmy, newdata = mydata))
corMat = cor(mydummifieddata)
corMasterList<- flattenSquareMatrix(cor.prob(mydummifieddata)) ## list of all correlations
## order by strength of correlation
corlist<- corMasterList[order(-abs(corMasterList$cor)),]
write.csv(corlist,paste0("FLAT correlation matrix_.csv"))
corlist <- corlist %>% dplyr::filter(j=="FPH_EtOH")   ## filter specific endpoint
head(corlist,50)
```

```
##                       i        j         cor            p
## 1               FPH_CO2 FPH_EtOH  1.00000000 0.000000e+00
## 2  calcCO2vols_increase FPH_EtOH  0.99236118 0.000000e+00
## 3                   ABW FPH_EtOH  0.91786994 0.000000e+00
## 4                   RDF FPH_EtOH  0.87954129 0.000000e+00
## 5                   ADF FPH_EtOH  0.87769061 0.000000e+00
## 6                    SG FPH_EtOH -0.86284096 0.000000e+00
## 7            daysonhops FPH_EtOH  0.79048699 0.000000e+00
## 8                    OE FPH_EtOH  0.42465387 1.825263e-06
## 9           variety.CIT FPH_EtOH -0.31130605 6.344556e-04
## 10           rouseFALSE FPH_EtOH -0.29912061 1.052549e-03
## 11            rouseTRUE FPH_EtOH  0.29912061 1.052549e-03
## 12         DH_temp.cold FPH_EtOH -0.28771356 1.658186e-03
## 13         DH_temp.warm FPH_EtOH  0.28771356 1.658186e-03
## 14         variety.CENT FPH_EtOH  0.19684854 3.340001e-02
## 15           pounds_bbl FPH_EtOH -0.18106806 5.073634e-02
## 16    form_of_hops.cone FPH_EtOH -0.13482145 1.472579e-01
## 17          variety.SIM FPH_EtOH -0.12741247 1.710099e-01
## 18  form_of_hops.ground FPH_EtOH  0.11346390 2.232010e-01
## 19         variety.AMAR FPH_EtOH  0.04549371 6.262168e-01
## 20         variety.CASC FPH_EtOH  0.03286017 7.250513e-01
## 21  form_of_hops.pellet FPH_EtOH  0.01530479 8.699053e-01
```

pairs.panels correlation matrix from library(psych)

```
## specify interesting variables:
interestingvariables<-c("ABW", "pounds_bbl", "daysonhops", "calcCO2vols_increase")
pairs.panels(mydummifieddata[c(interestingvariables, "FPH_EtOH")])
```



compare linear models

```
df <- FPHcalc
lm1<-lm(FPH_EtOH~daysonhops, data=df)
lm2<-lm(FPH_EtOH~daysonhops*form_of_hops, data=df)
lm3<-lm(FPH_EtOH~daysonhops*rouse, data=df)
lm4<-lm(FPH_EtOH~daysonhops*form_of_hops*rouse, data=df)

mtable1234 <- mtable("Model 1"=lm1,"Model 2"=lm2,"Model 3"=lm3, "Model 4"=lm4,
                     summary.stats=c("sigma","R-squared","F","p","N"),show.eqnames=T)
mtable1234b <- relabel(mtable1234,
                       "(Intercept)" = "Constant",
                       SG = "Specific Gravity",
                       ABW = "ABW = Ethanol (w/w)",
                       Er = "Er = Residual Extract (g/100mL)"
                       )
mtable1234
```

```
##
## Calls:
## Model 1: lm(formula = FPH_EtOH ~ daysonhops, data = df)
## Model 2: lm(formula = FPH_EtOH ~ daysonhops * form_of_hops, data = df)
## Model 3: lm(formula = FPH_EtOH ~ daysonhops * rouse, data = df)
## Model 4: lm(formula = FPH_EtOH ~ daysonhops * form_of_hops * rouse, data = df)
```

```
##
## ============================================================================
##                               Model 1     Model 2     Model 3     Model 4
##                             ----------- ----------- ----------- -----------
##                               FPH_EtOH    FPH_EtOH    FPH_EtOH    FPH_EtOH
## ----------------------------------------------------------------------------
##   (Intercept)                  0.549***    0.319       0.571***    0.329
##                               (0.045)     (0.190)     (0.050)     (0.191)
##   daysonhops                   0.041***    0.042***    0.039***    0.040***
##                               (0.003)     (0.003)     (0.004)     (0.004)
##   form_of_hops: ground/cone                0.849**                 0.849**
##                                           (0.268)                 (0.269)
##   form_of_hops: pellet/cone                0.214                   0.224
##                                           (0.193)                 (0.194)
##   rouse                                                -0.125      -0.107
##                                                       (0.140)     (0.134)
##   daysonhops x rouseTRUE                               0.007       0.006
##                                                       (0.007)     (0.007)
## ----------------------------------------------------------------------------
##   sigma                        0.343       0.328       0.344       0.330
##   R-squared                    0.625       0.662       0.628       0.665
##   F                          191.560      73.709      63.699      44.002
##   p                            0.000       0.000       0.000       0.000
##   N                          117         117         117         117
## ============================================================================
```

```
#show_html(mtable1234b)
```

modeling FPH

To make our model of the *Freshening Power of* **Centennial** *Hops*, we will narrow down the dataset as follows: *filter only include variety=="CENT"* remove the "rouse" samples (rousing did not significantly impact FPH) *create some "anchor rows" for time zero, to tell our model that there was zero ABW increase the moment hops were added. Even if we did have time zero data (we don't in most cases), instrument noise would likely cause extreme values in calculations and be problematic for modeling purposes.

```
#time zero rows "anchor rows"
## since we know for certain that at time zero, the true dry-hop induced ethanol increase is zero.  Some
## we'll make three identical rows per
unique_rows <- !duplicated(FPHcalc[c("variety","OX","harvestYear", "form_of_hops", "DH_temp")])  ## iden
df <- FPHcalc[unique_rows,]     ## subset unique combinations
df$daysonhops<- 0
df$timeofaddition<- 0
df$ABV<- 0
df$ABW<- 0
df$OE<- 0
df$Er<- 0
df$Ea<- 0
df$SG<- 0
df$RDF<- 0
df$ADF<- 0
df$dhop_day<- 0
df$contact_days<- 0
```

```
df$ABV_increase<- 0
df$ABW.REF_NH<- 0
df$ABW_increase<- 0
df$FPH_EtOH<- 0
df$FPH_CO2<- 0
timezerorows<-rbind(df,df)    ### duplicates
timezerorows<-rbind(timezerorows,df)  ### triplicates
df2<-rbind(FPHcalc,timezerorows)  ### triplicates
```

narrow down dataset and compare linear models

```
## narrow down dataset
df<- df2 %>% dplyr::filter(variety=="CENT"&rouse==FALSE&DH_temp=="warm")      # filter rows

df<- df %>% dplyr::select(FPH_EtOH,daysonhops,brew_date,form_of_hops)  # select columns


#compare models

lm1<-lm(FPH_EtOH~daysonhops, data=df)
lm2<-lm(FPH_EtOH~daysonhops*form_of_hops, data=df)
lm3<-lm(FPH_EtOH~daysonhops*brew_date, data=df)
lm4<-lm(FPH_EtOH~daysonhops*brew_date*form_of_hops, data=df)

mtable1234 <- mtable("Model 1"=lm1,"Model 2"=lm2,"Model 3"=lm3, "Model 4"=lm4,
                     summary.stats=c("sigma","R-squared","F","p","N"),show.eqnames=T)
mtable1234b <- relabel(mtable1234,
                       "(Intercept)" = "Constant",
                       SG = "Specific Gravity",
                       ABW = "ABW = Ethanol (w/w)",
                       Er = "Er = Residual Extract (g/100mL)"
                       )
mtable1234
```

```
##
## Calls:
## Model 1: lm(formula = FPH_EtOH ~ daysonhops, data = df)
## Model 2: lm(formula = FPH_EtOH ~ daysonhops * form_of_hops, data = df)
## Model 3: lm(formula = FPH_EtOH ~ daysonhops * brew_date, data = df)
## Model 4: lm(formula = FPH_EtOH ~ daysonhops * brew_date * form_of_hops,
##     data = df)
##
## ============================================================================
##                                    Model 1    Model 2    Model 3    Model 4
##                                   ---------- --------- ---------- ----------
##                                    FPH_EtOH   FPH_EtOH  FPH_EtOH   FPH_EtOH
## ----------------------------------------------------------------------------
##   (Intercept)                       0.378***   0.000     0.000      0.000
##                                    (0.048)    (0.196)   (0.113)    (0.137)
##   daysonhops                        0.050***   0.105     0.108***  -0.003
##                                    (0.004)    (0.055)   (0.027)    (0.036)
##   form_of_hops: ground/cone                   -0.000               -0.000
```

```
##                                                (0.277)             (0.130)
##   form_of_hops: pellet/cone                     0.393              -0.000
##                                                (0.202)             (0.101)
##   daysonhops x form_of_hopsground               0.170*              0.170***
##                                                (0.078)             (0.037)
##   daysonhops x form_of_hopspellet              -0.057               0.112***
##                                                (0.056)             (0.028)
##   brew_date: 2017-07-23/2017-07-02                          0.485***   0.485***
##                                                            (0.135)    (0.110)
##   brew_date: 2017-08-06/2017-07-02                         -0.127     -0.127
##                                                            (0.160)    (0.130)
##   brew_date: 2017-08-29/2017-07-02                         -0.423**   -0.423**
##                                                            (0.160)    (0.130)
##   brew_date: 2017-10-06/2017-07-02                         -0.000     -0.000
##                                                            (0.121)    (0.101)
##   brew_date: 2017-10-30/2017-07-02                          0.177      0.177
##                                                            (0.163)    (0.133)
##   daysonhops x brew_date2017-07-23                         -0.067*    -0.067**
##                                                            (0.027)    (0.022)
##   daysonhops x brew_date2017-10-06                          0.102***   0.109***
##                                                            (0.029)    (0.024)
##   daysonhops x brew_date2017-10-30                         -0.002     -0.002
##                                                            (0.032)    (0.026)
## ------------------------------------------------------------------------------
##   sigma                                 0.372      0.340     0.196      0.160
##   R-squared                             0.632      0.707     0.907      0.941
##   F                                   156.167     42.039    90.388     97.389
##   p                                     0.000      0.000     0.000      0.000
##   N                                    93         93        93         93
## ==============================================================================
```

```
#show_html(mtable1234b)
```

The R Book (Crawley) Table 20.1: nonlinear functions useful in biology

Table 20.1. Useful non-linear functions EXPANDED:

Function Class	name	equation	example code	example applications
Asymptotic functions	Michaelis–Menten	$y = \frac{ax}{1+bx}$	nls(bone~a*age/(1+b*age),start=list(a=8,b=0.08)) nls(rate~SSmicmen(conc,a,b))	enzyme reactions bird
	2-parameter asymptotic exponential	$y = a(1e^{bx})$	nls(bone~a*(1-exp(-c*age)),start=list(a=120,c=0.064))	(1-

Function Class	name	equation	example code	example applications
	3-parameter asymptotic exponential	$y = abe^{cx}$	nls(bone~a-b*exp(-c*age),start=list(a=120,b=110,c=0.064))	tbd
			nls(bone~SSasy	Sasymp(age,a,b,c))
			nls(density ~ SSlogis(log(concentration), a, b, c))	tbd
S-shaped functions	2-parameter logistic	$y = \frac{e^{a+bx}}{1+e^{a+bx}}$		tbd
	3-parameter logistic	$y = \frac{a}{1+be^{cx}}$		tbd
	4-parameter logistic	$y = a + \frac{b-a}{1+e^{(cx)/d}}$	nls(weight~SSf	pl(Time, a, b, c, d))
	Weibull	$y = abe^{(cx^d)}$	nls(weight ~ SSweibull(time, Asym, Drop, lrc, pwr))	tbd
	Gompertz	$y = ae^{be^{cx}}$		tbd
Humped curves	Ricker curve	$y = axe^{bx}$		tbd
	First-order compartment	$y = kexp(exp(a)x)Time(exp(b)x)$	nls(conc~SSfol	lod(Dose, Time, exp(a), exp(b), b, c))
	Bell-shaped	$y = aexp(ABS(bx)^2)$		tbd
	Biexponential	$y = ae^{bx}ce^{dx}$		tbd

```
sessionInfo()
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 17134)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
```

```
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] bindrcpp_0.2.2    officer_0.3.2     caret_6.0-81
##  [4] psych_1.8.4       memisc_0.99.14.12 MASS_7.3-51.1
##  [7] lattice_0.20-38   gridExtra_2.3     ggplot2_3.0.0
## [10] kableExtra_1.0.1  magrittr_1.5      flextable_0.4.6
## [13] knitr_1.21        dplyr_0.7.4
##
## loaded via a namespace (and not attached):
##  [1] httr_1.3.1          jsonlite_1.5       viridisLite_0.3.0
##  [4] splines_3.5.2       foreach_1.4.4      prodlim_2018.04.18
##  [7] assertthat_0.2.0    stats4_3.5.2       yaml_2.1.19
## [10] gdtools_0.1.7       ipred_0.9-8        pillar_1.2.2
## [13] backports_1.1.2     glue_1.2.0         uuid_0.1-2
## [16] digest_0.6.15       rvest_0.3.2        colorspace_1.3-2
## [19] recipes_0.1.4       htmltools_0.3.6    Matrix_1.2-15
## [22] plyr_1.8.4          timeDate_3043.102  pkgconfig_2.0.1
## [25] purrr_0.2.4         scales_0.5.0       webshot_0.5.1
## [28] gower_0.1.2         lava_1.6.4         tibble_1.4.2
## [31] generics_0.0.2      withr_2.1.2        repr_0.19.1
## [34] nnet_7.3-12         lazyeval_0.2.1     cli_1.0.0
## [37] mnormt_1.5-5        crayon_1.3.4       survival_2.43-3
## [40] evaluate_0.10.1     nlme_3.1-137       xml2_1.2.0
## [43] foreign_0.8-71      class_7.3-14       tools_3.5.2
## [46] data.table_1.12.0   hms_0.4.2          stringr_1.3.1
## [49] munsell_0.4.3       zip_1.0.0          compiler_3.5.2
## [52] rlang_0.3.1         grid_3.5.2         iterators_1.0.10
## [55] rstudioapi_0.7      labeling_0.3       base64enc_0.1-3
## [58] rmarkdown_1.9       gtable_0.2.0       ModelMetrics_1.2.2
## [61] codetools_0.2-15    reshape2_1.4.3     R6_2.2.2
## [64] lubridate_1.7.4     utf8_1.1.3         bindr_0.1.1
## [67] rprojroot_1.3-2     readr_1.1.1        stringi_1.1.7
## [70] parallel_3.5.2      Rcpp_0.12.16       rpart_4.1-13
## [73] tidyselect_0.2.4    xfun_0.4
```