Beverage Review Project

Nick McCulloch

Table of Contents

# Introduction

## Project Description

The purpose of this project was to better understand factors that contribute to consumer ratings for particular beverages, namely, beer. The beer industry is booming in the American market, with consumers switching slowly from drinks with a high alcohol percentage to lighter and more refreshing beverages. Consumer trends can be seen in overall beverage ratings and in ratings for subcategories such as palate, aroma, or Alcohol By Volume. By focusing on these essential categories, key predictors can be identified and used to direct further development.

## Data Description

The initial dataset consists of 1.5 million consumer reviews provided by BeerAdvocate.com, a site where “beer geeks and industry professionals” can submit numeric and written reviews on a wide array of beverages. The focus of the project was larger trends in consumer preferences accross products so this data was filtered to exclude products with fewer than 500 reviews, aggregated by style and then filtered again to exclude any style with fewer than 5 products. The resulting dataset consisted of 25 styles, 275 individual products, and 270,323 individual consumer reviews.

The variable of interest is the Overall Score of a drink on a continuous scale from 0 to 5, with a score of 5.0 being the highest.

The key predictors were Beer Style Category (BSC), Review Count, Alcohol by Volume (ABV), Aroma Score, Appearance Score, Palate Score, and Taste Score.

## Packages

library(pacman)  
  
p\_load(stats, lmtest, dplyr, klaR, car, glmnet, pls, ggplot2, patchwork, ggthemes, reshape2)

## Reading Data

# Reading in the data with strings as factors b/c of text in style column  
beer\_data <- read.table("beer\_data.csv", header = T, sep = ",", stringsAsFactors = T)  
  
# Checking the data size, columns, class, etc.  
head(beer\_data)

## style abv appearance aroma overall palate taste revcount  
## 1 American Adjunct Lager 4.60 2.29 1.95 2.61 2.30 2.19 1096  
## 2 American Adjunct Lager 4.60 2.65 2.42 3.09 2.72 2.64 712  
## 3 American Adjunct Lager 4.70 2.99 2.76 3.37 3.03 3.05 894  
## 4 American Adjunct Lager 4.80 2.73 2.58 3.08 2.87 2.84 520  
## 5 American Adjunct Lager 5.00 2.68 2.45 3.11 2.74 2.76 545  
## 6 American Adjunct Lager 4.74 2.90 2.68 3.61 3.01 3.10 1460

names(beer\_data)

## [1] "style" "abv" "appearance" "aroma" "overall"   
## [6] "palate" "taste" "revcount"

str(beer\_data)

## 'data.frame': 275 obs. of 8 variables:  
## $ style : Factor w/ 25 levels "American Adjunct Lager",..: 1 1 1 1 1 1 2 2 2 2 ...  
## $ abv : num 4.6 4.6 4.7 4.8 5 4.74 5.6 5.2 5.8 6 ...  
## $ appearance: num 2.29 2.65 2.99 2.73 2.68 2.9 3.96 3.73 3.81 4.03 ...  
## $ aroma : num 1.95 2.42 2.76 2.58 2.45 2.68 3.81 3.45 3.66 3.8 ...  
## $ overall : num 2.61 3.09 3.37 3.08 3.11 3.61 3.98 3.82 4.03 3.81 ...  
## $ palate : num 2.3 2.72 3.03 2.87 2.74 3.01 3.85 3.57 3.84 3.76 ...  
## $ taste : num 2.19 2.64 3.05 2.84 2.76 3.1 3.91 3.6 3.91 3.81 ...  
## $ revcount : int 1096 712 894 520 545 1460 723 1675 765 1117 ...

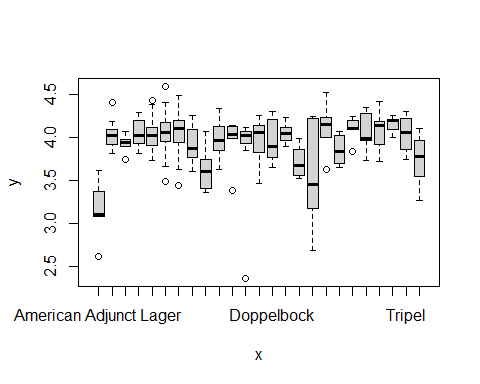
data\_summary <- summary(beer\_data)

The number of reviews per product and ABV had the highest variance among predictors. The median number of reviews per product is 772 and the range is 2,500 and a standard deviation (SD) of 510.62. ABV had a median of 7.10, a range of 13.50, and an SD of 2.42. All the other predictors have medians between 3.99 (review taste) and 4.07 (review appearance) with ranges from 2.23 (review overall) to 2.62 (review aroma). The overall review score has a mean of 3.962, a median of 4.02, and a range of 2.253. Considering that the maximum overall review score is out of 5, the range of scores varies by as much as 45% of the highest possible score. With the exception of review count and ABV, all the other predictors show strong negative skewness (to the left) with values from -1.43 (review palate) to -1.92 (review appearance). All predictors show positive kurtosis implying there are more extreme values in each predictor than in a normal distribution and have fatter tails or a leptokurtic distribution. Review appearance has the highest level of positive kurtosis (6.28) followed by review overall with 5.53.

# Preliminary Tests and Transformations

## Test Plot

# quick check with base R plot to get sense of data.  
plot(beer\_data$style, beer\_data$overall)

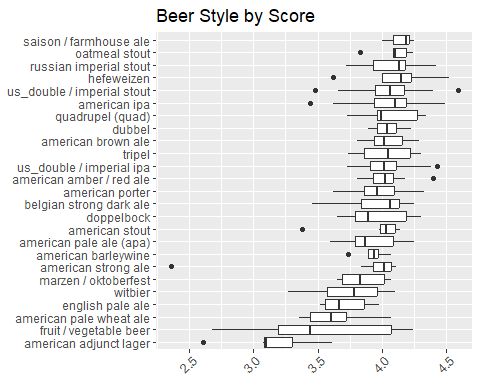


## Transformations

# transforming 'styles' strings to lower case to improve spacing  
  
beer\_data$style <- tolower(beer\_data$style)  
  
# replacing american double to shorten string  
  
beer\_data$style <- gsub("american double", "us\_double", beer\_data$style)  
  
beer\_data$style <- factor(beer\_data$style)

## Improved Test Plot

# Now an improved boxplot  
a\_data <- ggplot(beer\_data, aes(x = overall, y = reorder(style, overall)))  
  
a\_graph <- geom\_boxplot()  
  
a\_theme <- theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust = 1, margin = margin(t = 1, r = 20, b = 1, l = 0)))  
  
a\_labs <- labs(title = "Beer Style by Score", x = NULL, y = NULL)  
  
#a\_scale <- scale\_y\_discrete(guide = guide\_axis(check.overlap = T))  
  
a\_data + a\_graph + a\_theme + a\_labs #+ a\_scale



## Transformations

## Basic Linear Model

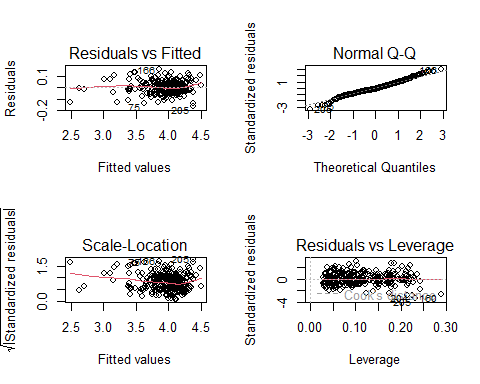
lm\_basic <- lm(overall ~ ., data = beer\_data)  
  
basic\_summary <- summary(lm\_basic)  
  
basic\_summary

##   
## Call:  
## lm(formula = overall ~ ., data = beer\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.162203 -0.030640 -0.006417 0.033219 0.157004   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.289e-01 6.545e-02 12.664 < 2e-16 \*\*\*  
## styleamerican amber / red ale -5.232e-02 4.359e-02 -1.200 0.231233   
## styleamerican barleywine -1.319e-01 5.210e-02 -2.531 0.011996 \*   
## styleamerican brown ale -1.188e-01 4.538e-02 -2.617 0.009415 \*\*   
## styleamerican ipa -3.096e-02 4.299e-02 -0.720 0.472106   
## styleamerican pale ale (apa) -2.709e-02 3.994e-02 -0.678 0.498192   
## styleamerican pale wheat ale 2.787e-02 4.080e-02 0.683 0.495240   
## styleamerican porter -1.244e-01 4.292e-02 -2.898 0.004096 \*\*   
## styleamerican stout -1.559e-01 4.801e-02 -3.247 0.001330 \*\*   
## styleamerican strong ale -1.656e-01 4.713e-02 -3.514 0.000527 \*\*\*  
## stylebelgian strong dark ale -9.727e-02 4.587e-02 -2.121 0.034969 \*   
## styledoppelbock -1.528e-01 4.471e-02 -3.418 0.000738 \*\*\*  
## styledubbel -9.698e-02 4.773e-02 -2.032 0.043252 \*   
## styleenglish pale ale -2.688e-02 4.229e-02 -0.636 0.525655   
## stylefruit / vegetable beer -1.780e-01 4.692e-02 -3.794 0.000187 \*\*\*  
## stylehefeweizen -7.998e-03 4.374e-02 -0.183 0.855076   
## stylemarzen / oktoberfest -6.736e-02 4.405e-02 -1.529 0.127531   
## styleoatmeal stout -1.466e-01 4.938e-02 -2.970 0.003278 \*\*   
## stylequadrupel (quad) -8.563e-02 5.079e-02 -1.686 0.093111 .   
## stylerussian imperial stout -8.295e-02 4.799e-02 -1.728 0.085186 .   
## stylesaison / farmhouse ale -2.733e-02 4.845e-02 -0.564 0.573176   
## styletripel -3.673e-02 4.764e-02 -0.771 0.441432   
## styleus\_double / imperial ipa -6.176e-02 4.705e-02 -1.313 0.190510   
## styleus\_double / imperial stout -1.073e-01 4.825e-02 -2.224 0.027043 \*   
## stylewitbier -2.194e-02 4.050e-02 -0.542 0.588521   
## abv -3.635e-02 2.649e-03 -13.725 < 2e-16 \*\*\*  
## appearance 2.800e-03 4.277e-02 0.065 0.947865   
## aroma -3.361e-01 5.143e-02 -6.535 3.69e-10 \*\*\*  
## palate -7.886e-02 8.913e-02 -0.885 0.377173   
## taste 1.286e+00 8.645e-02 14.872 < 2e-16 \*\*\*  
## revcount -2.470e-05 7.443e-06 -3.318 0.001044 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.05586 on 244 degrees of freedom  
## Multiple R-squared: 0.9681, Adjusted R-squared: 0.9641   
## F-statistic: 246.6 on 30 and 244 DF, p-value: < 2.2e-16

tempsymbol1 Checking Assumptions for OLS

## Residual Plots and Breusch-Pagan

par(mfrow = c(2,2))  
plot(lm\_basic)



The residuals/fitted plot isn’t perfect with the largest deviation from linearity to the right. The QQ plot shows mildly fat tails, indicating more data located at the extremes. The scale location plot is neither horizontal nor evenly spread, indicating heteroskedasticty. The residuals vs leverage plot does indicate some notable outliers but none that fall past the Cook’s distance.

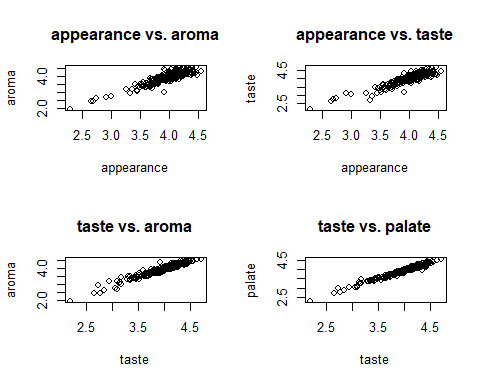
bptest(lm\_basic, data = beer\_data)

##   
## studentized Breusch-Pagan test  
##   
## data: lm\_basic  
## BP = 73.433, df = 30, p-value = 1.66e-05

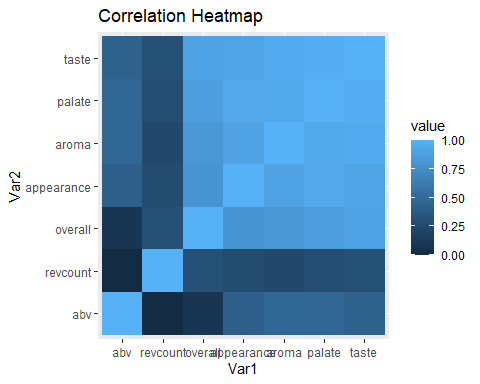
The BP test strongly suggests heteroskedasticity.

## Visual Check for Multicollinearity

par(mfrow = c(2,2))  
  
attach(beer\_data)  
plot(appearance, aroma, main="appearance vs. aroma")  
plot(appearance,taste, main="appearance vs. taste")  
plot(taste, aroma, main="taste vs. aroma")  
plot(taste, palate, main="taste vs. palate")

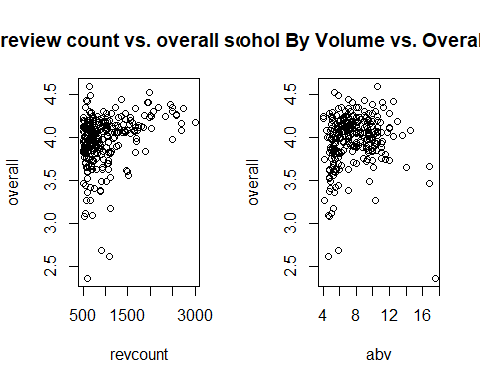


#excluding 'style' from the corr plot, which is the only non-numeric column  
corr\_data <- beer\_data[-1]  
  
# creating correlation matrix  
corr\_mat <- round(cor(corr\_data),2)  
   
# sorting matrix for easier interpretation  
dist <- as.dist((1-corr\_mat)/2)  
   
# clustering the dist matrix  
hclust <- hclust(dist)  
corr\_mat <- corr\_mat[hclust$order, hclust$order]  
   
# reduce the size of correlation matrix  
melted\_corr\_mat <- melt(corr\_mat)  
   
#plotting the correlation heatmap  
ggplot(data = melted\_corr\_mat, aes(x = Var1, y = Var2, fill = value)) +  
geom\_tile() + labs(title = "Correlation Heatmap")



Somewhat unsurprisingly, similar or related categories in the data like ‘taste’ and ‘aroma’ seem to be correlated. The plots above demonstrate the severe multicollinearity found in the data.

par(mfrow = c(1,2))  
  
plot(revcount, overall, main="review count vs. overall score")  
plot(abv, overall, main="Alcohol By Volume vs. Overall Score")



There was not a similar level of correlation observed between overall score and ABV or number of reviews.

# Main Analysis and Model Testing

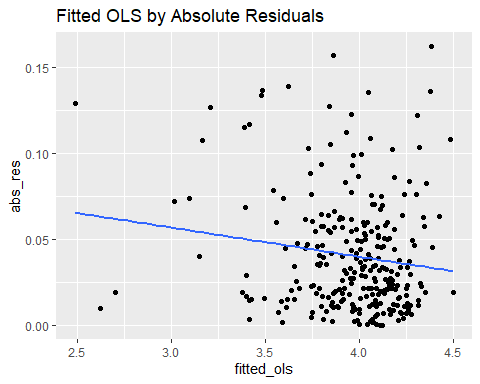
## Weighted Least Squares (WLS)

The multicollinearity and heteroskedasticity found above suggested the need to test a WLS model, and because the residuals have a Gaussian distribution, a GLM model, as well.

# First extracting fitted values from the model to create weights  
fitted\_ols <- fitted(lm\_basic)  
  
abs\_res <- abs(residuals(lm\_basic))  
  
cbind(fitted\_ols, abs\_res)[1:10, ]

## fitted\_ols abs\_res  
## 1 2.619858 0.009858182  
## 2 3.017822 0.072178062  
## 3 3.399052 0.029051686  
## 4 3.207049 0.127048631  
## 5 3.150109 0.040108906  
## 6 3.476111 0.133889343  
## 7 4.009083 0.029082897  
## 8 3.743978 0.076021811  
## 9 4.051553 0.021553303  
## 10 3.866900 0.056900035

lm\_abs\_res <- lm(abs\_res ~ fitted\_ols)  
  
# sanity check  
# fitted(lm\_abs\_res)[1:10]  
  
fit\_df<-as.data.frame(cbind(fitted\_ols, abs\_res))  
  
# plotting the values  
ggplot(data = fit\_df, aes(x = fitted\_ols, y = abs\_res))+  
 geom\_point()+  
 stat\_smooth(method ="lm", se = FALSE)+  
 labs(title = "Fitted OLS by Absolute Residuals")



# saving the weights to new variable  
wts <- 1 / fitted(lm\_abs\_res)^2  
  
# fiting WLS and WGLM models using the weights  
fit\_wls <- lm(overall ~ ., data = beer\_data, weights = wts)  
  
fit\_wglm <- glm(overall ~ ., data = beer\_data, weights = wts)  
  
# creating summaries of the new models for comparison  
sum\_lm\_basic <- summary(lm\_basic) # R-squared 0.9641  
sum\_fit\_wglm <- summary(fit\_wglm)   
sum\_fit\_wls <- summary(fit\_wls) # R-squared - 0.9548  
  
paste("lm basic-adjusted r-squared: ", format(sum\_lm\_basic$adj.r.squared, digits = 4))

## [1] "lm basic-adjusted r-squared: 0.9641"

paste("fit\_wls-adjusted r-squared: ", format(sum\_fit\_wls$adj.r.squared, digits = 4))

## [1] "fit\_wls-adjusted r-squared: 0.9548"

## WGLM

# first running regression with exisiting levels  
  
beer\_fit <- glm(overall ~ ., data = beer\_data)  
  
basic\_sum <- summary(lm\_basic)  
  
fit\_sum <- summary(beer\_fit)  
  
# extracting the coefficients, droppping non-style coefs, reordering, and rounding  
# could have used coef(summary(x)) instead.  
basic\_coef <- data.frame(basic\_sum$coefficients)  
basic\_coef <- basic\_coef[!(row.names(basic\_coef) %in% c("abv","appearance","aroma","palate","taste","revcount","(Intercept)")),]  
basic\_coef <- round(basic\_coef, 4)  
basic\_coef$coefficients <- rownames(basic\_coef)  
basic\_coef <- basic\_coef[, c(5, 1, 2, 3, 4)]  
  
fit\_coef <- data.frame(fit\_sum$coefficients)  
fit\_coef <- fit\_coef[!(row.names(fit\_coef) %in% c("abv","appearance","aroma","palate","taste","revcount", "(Intercept)")),]  
fit\_coef <- round(fit\_coef, 4)  
fit\_coef$coefficients <- rownames(fit\_coef)  
fit\_coef <- fit\_coef[, c(5, 1, 2, 3, 4)]  
  
# extracting style with highest p-value and lowest estimate  
attach(basic\_coef)  
level\_data<- rbind(basic\_coef[Pr...t.. == max(Pr...t..),],  
 basic\_coef[Estimate == max(Estimate),])  
detach(basic\_coef)  
attach(fit\_coef)  
level\_data<- rbind(level\_data, fit\_coef[Pr...t.. == max(Pr...t..),],  
 fit\_coef[Estimate == max(Estimate),])  
rownames(level\_data) <- c("basic coef-highest p-value:", "basic coef-lowest estimate:", "fit\_coef-highest p-value:", "fit\_coef-lowest estimate:")  
detach(fit\_coef)  
  
print(level\_data[,c(1,2,5)])

## coefficients Estimate Pr...t..  
## basic coef-highest p-value: stylehefeweizen -0.0080 0.8551  
## basic coef-lowest estimate: styleamerican pale wheat ale 0.0279 0.4952  
## fit\_coef-highest p-value: stylehefeweizen -0.0080 0.8551  
## fit\_coef-lowest estimate: styleamerican pale wheat ale 0.0279 0.4952

‘American Pale Wheat Ale’ has the lowest estimate and ‘Hefeweizen’ turns out to have the highest p-value value, making either a good candiate for a releveled reference.

## Releveled WGLM

beer\_datarlv <- beer\_data  
  
# releveling the factors  
beer\_datarlv$style <- relevel(beer\_datarlv$style, ref = "american pale wheat ale")  
  
# sanity check  
#levels(beer\_datarlv$style)  
  
# fitting releveld model  
beer\_rlv <- glm(overall ~., data = beer\_datarlv)

# anova table for the new model.  
anova(beer\_rlv)

## Analysis of Deviance Table  
##   
## Model: gaussian, link: identity  
##   
## Response: overall  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev  
## NULL 274 23.8426  
## style 24 8.6118 250 15.2307  
## abv 1 1.0856 249 14.1451  
## appearance 1 8.8204 248 5.3247  
## aroma 1 2.6609 247 2.6638  
## palate 1 1.2114 246 1.4524  
## taste 1 0.6568 245 0.7956  
## revcount 1 0.0344 244 0.7613

# creating more readable summary for the coefficients  
rlv\_summ <- summary(beer\_rlv)  
  
rlv\_coefs <- data.frame(rlv\_summ$coefficients)  
  
rlv\_coefs <- rlv\_coefs[order(rlv\_coefs$Estimate),]  
  
pv <- rlv\_coefs[, "Pr...t.."]  
  
signif <- symnum(pv, corr = FALSE, na = FALSE,   
 cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),   
 symbols = c("\*\*\*", "\*\*", "\*", ".", " "))  
  
legend <- attr(signif,"legend")  
  
signif <- as.vector(signif)  
  
rlv\_coefs<-cbind(rlv\_coefs,signif)  
  
#[1] 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  
  
rlv\_coefs <- rlv\_coefs[,1:5]  
rlv\_coefs

## Estimate Std..Error t.value  
## aroma -3.360627e-01 5.142872e-02 -6.5345348  
## stylefruit / vegetable beer -2.058804e-01 3.463372e-02 -5.9445070  
## styleamerican strong ale -1.934559e-01 3.434261e-02 -5.6331153  
## styleamerican stout -1.837495e-01 3.508303e-02 -5.2375614  
## styledoppelbock -1.807029e-01 3.338725e-02 -5.4123335  
## styleoatmeal stout -1.744986e-01 3.718713e-02 -4.6924466  
## styleamerican barleywine -1.597382e-01 3.885815e-02 -4.1108039  
## styleamerican porter -1.522432e-01 3.151962e-02 -4.8301082  
## styleamerican brown ale -1.466514e-01 3.499230e-02 -4.1909632  
## styleus\_double / imperial stout -1.351809e-01 3.283286e-02 -4.1172430  
## stylebelgian strong dark ale -1.251380e-01 3.397143e-02 -3.6836241  
## styledubbel -1.248462e-01 3.486655e-02 -3.5806865  
## stylequadrupel (quad) -1.134913e-01 3.872681e-02 -2.9305614  
## stylerussian imperial stout -1.108148e-01 3.310909e-02 -3.3469598  
## stylemarzen / oktoberfest -9.522087e-02 3.512511e-02 -2.7109061  
## styleus\_double / imperial ipa -8.962495e-02 3.171264e-02 -2.8261589  
## styleamerican amber / red ale -8.018481e-02 3.062467e-02 -2.6183079  
## palate -7.885975e-02 8.913415e-02 -0.8847310  
## styletripel -6.459610e-02 3.475638e-02 -1.8585392  
## styleamerican ipa -5.882556e-02 2.805055e-02 -2.0971268  
## stylesaison / farmhouse ale -5.519622e-02 3.445462e-02 -1.6019978  
## styleamerican pale ale (apa) -5.495697e-02 2.851739e-02 -1.9271391  
## styleenglish pale ale -5.474109e-02 3.181933e-02 -1.7203721  
## stylewitbier -4.980433e-02 2.896779e-02 -1.7193005  
## abv -3.635137e-02 2.648640e-03 -13.7245416  
## stylehefeweizen -3.586350e-02 3.110547e-02 -1.1529642  
## styleamerican adjunct lager -2.786577e-02 4.079765e-02 -0.6830239  
## revcount -2.469903e-05 7.443395e-06 -3.3182475  
## appearance 2.799792e-03 4.277417e-02 0.0654552  
## (Intercept) 8.567276e-01 8.615629e-02 9.9438781  
## taste 1.285649e+00 8.644546e-02 14.8723702  
## Pr...t.. signif  
## aroma 3.690529e-10 \*\*\*  
## stylefruit / vegetable beer 9.531555e-09 \*\*\*  
## styleamerican strong ale 4.861443e-08 \*\*\*  
## styleamerican stout 3.514852e-07 \*\*\*  
## styledoppelbock 1.485610e-07 \*\*\*  
## styleoatmeal stout 4.501026e-06 \*\*\*  
## styleamerican barleywine 5.389420e-05 \*\*\*  
## styleamerican porter 2.411482e-06 \*\*\*  
## styleamerican brown ale 3.886005e-05 \*\*\*  
## styleus\_double / imperial stout 5.250625e-05 \*\*\*  
## stylebelgian strong dark ale 2.832080e-04 \*\*\*  
## styledubbel 4.134643e-04 \*\*\*  
## stylequadrupel (quad) 3.704478e-03 \*\*  
## stylerussian imperial stout 9.460599e-04 \*\*\*  
## stylemarzen / oktoberfest 7.186604e-03 \*\*  
## styleus\_double / imperial ipa 5.101262e-03 \*\*  
## styleamerican amber / red ale 9.389564e-03 \*\*  
## palate 3.771727e-01   
## styletripel 6.429665e-02 .  
## styleamerican ipa 3.701185e-02 \*  
## stylesaison / farmhouse ale 1.104499e-01   
## styleamerican pale ale (apa) 5.512341e-02 .  
## styleenglish pale ale 8.663263e-02 .  
## stylewitbier 8.682785e-02 .  
## abv 3.749755e-32 \*\*\*  
## stylehefeweizen 2.500533e-01   
## styleamerican adjunct lager 4.952397e-01   
## revcount 1.044149e-03 \*\*  
## appearance 9.478652e-01   
## (Intercept) 8.864799e-20 \*\*\*  
## taste 4.767141e-36 \*\*\*

## Cross-validation – comparing OLS to GLM

## Condition Index

# library(klaR)  
ci\_basic <- cond.index(lm\_basic, data = beer\_data)  
ci\_basic[length(ci\_basic)]

## [1] 405.164

ci\_glm <- cond.index(beer\_rlv, data = beer\_data)  
ci\_glm[length(ci\_glm)]

## [1] 405.164

The high condition index indicates both models suffer from severe multicollinearity (as indicated in the previous plots).

# fitting an ad-hoc model that excludes several of the correlated variables.  
new\_lm <- glm(overall ~ abv + appearance + taste + revcount + style, data = beer\_data)  
ci\_new <- cond.index(new\_lm, data = beer\_data)  
ci\_new[length(ci\_new)]

## [1] 119.9855

The model above excludes the ‘aroma’ and ‘palate’ variables which seemed to be highly correlated with ‘taste’. The resulting condition index was significantly improved.

## VIF

# library(car)  
print("lm\_basic\_vif")

## [1] "lm\_basic\_vif"

vif(lm\_basic)

## GVIF Df GVIF^(1/(2\*Df))  
## style 27.660328 24 1.071615  
## abv 3.605780 1 1.898889  
## appearance 15.281654 1 3.909176  
## aroma 32.365236 1 5.689045  
## palate 71.021952 1 8.427452  
## taste 87.562967 1 9.357509  
## revcount 1.268638 1 1.126338

print("new\_lm\_vif")

## [1] "new\_lm\_vif"

vif(new\_lm)

## GVIF Df GVIF^(1/(2\*Df))  
## abv 3.257379 1 1.804821  
## appearance 10.200315 1 3.193793  
## taste 7.358349 1 2.712628  
## revcount 1.237765 1 1.112549  
## style 10.889703 24 1.051004

Calculating the VIF for both models offers further support for the conclusion that there is severe multicollinearity. The initial model has VIF’s over 5 for ‘aroma’, ‘palate’, and ‘taste’. By contrast the reduced model, ‘lm\_new’, which excluded ‘aroma’ and ‘palate’ shows no variable with a VIF over 5. This suggests that variable selection to reduce the predictor variables would be helpful.

## Ridge Prep - Splitting and Training the Data

Ridge regression is a suitable method for variable selection but requires some preparation.

# creating training and test datasets  
RNGkind(sample.kind = "default")  
options(scipen = 4)  
  
paste("total rows: ",nrow(beer\_data))

## [1] "total rows: 275"

tr\_size <- 0.8  
train <- sample(nrow(beer\_data), tr\_size \* nrow(beer\_data))  
  
beer\_train <- beer\_data[train, ]  
paste("training data rows: ",nrow(beer\_train))

## [1] "training data rows: 220"

beer\_test <- beer\_data[-train, ]  
paste("test data rows: ", nrow(beer\_test))

## [1] "test data rows: 55"

paste(c("training data sample:", train[1:10]),collapse = " ")

## [1] "training data sample: 136 46 137 29 263 123 59 53 103 227"

# fitting model to training data  
  
fit\_train <- lm(overall ~ ., data = beer\_train)  
train\_mse <- mean(fit\_train$residuals^2)  
c("Train MSE" = train\_mse, "Train RMSE" = sqrt(train\_mse))

## Train MSE Train RMSE   
## 0.002742292 0.052366902

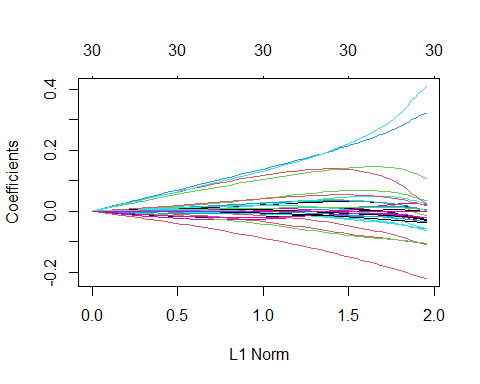
## Random Split Cross-Validation Test MSE

# fitting model to test data  
pred\_test <- predict(fit\_train, beer\_test)  
test\_mse\_rs <- mean((beer\_test$overall - pred\_test)^2)  
c("RSCV Test MSE" = test\_mse\_rs, "RSCV Test RMSE" = sqrt(test\_mse\_rs))

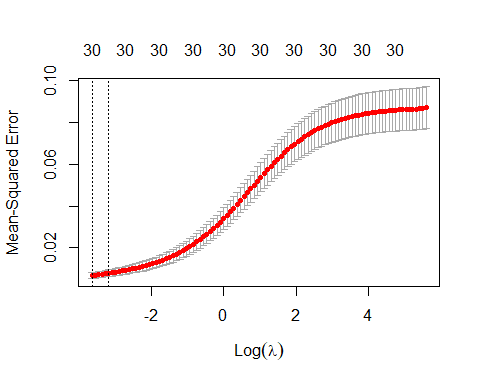
## RSCV Test MSE RSCV Test RMSE   
## 0.003245172 0.056966414

## Ridge

x <- model.matrix(overall ~ ., data = beer\_data)[, -1]  
y <- beer\_data$overall  
  
# first creating ridge reg. without specifying lambda  
ridge\_reg <- glmnet(x, y, alpha = 0)  
  
plot(ridge\_reg)



# now recreating with cv.glmnet to find optimal lambda  
ridge\_reg\_cv10 <- cv.glmnet(x, y, alpha = 0)  
  
lambda\_vec <- round(cbind("Lambda" = ridge\_reg\_cv10$lambda, "10FCV" = ridge\_reg\_cv10$cvm), digits = 3)  
  
plot(ridge\_reg\_cv10)



# the best lamdas can be seen here  
top\_5\_lambdas <- tail(lambda\_vec)  
#top\_5\_lambdas  
  
# finding best lambda and best cv\_ridge  
ridge\_best\_lambda <- ridge\_reg\_cv10$lambda.min  
  
min\_cv\_ridge <- min(ridge\_reg\_cv10$cvm)

round(cbind("Best Lambda" = ridge\_best\_lambda, "Best 10FCV" = min\_cv\_ridge), digits = 3)

## Best Lambda Best 10FCV  
## [1,] 0.027 0.007

best\_model <- glmnet(x, y, alpha = 0, lambda = ridge\_best\_lambda)  
  
# tidying data for easier interpretation  
best\_matrix <- as.matrix(coef(best\_model))  
best\_matrix <- cbind(row.names(best\_matrix),best\_matrix)  
best\_matrix <- data.frame(best\_matrix)  
best\_matrix <- best\_matrix[order(best\_matrix$s0),]  
colnames(best\_matrix) <- c("variable", "coef")  
rownames(best\_matrix) <- 1:nrow(best\_matrix)  
best\_matrix

## variable coef  
## 1 styleenglish pale ale -0.00147983715333223  
## 2 styleamerican brown ale -0.0135474023380459  
## 3 stylewitbier -0.0207022679830009  
## 4 stylequadrupel (quad) -0.025593594197431  
## 5 stylebelgian strong dark ale -0.025870170917011  
## 6 stylerussian imperial stout -0.0259590839340409  
## 7 styleamerican porter -0.0306600069823958  
## 8 styleus\_double / imperial ipa -0.0335898139871095  
## 9 styledubbel -0.034973305970971  
## 10 abv -0.0369243893344119  
## 11 styleoatmeal stout -0.0563000466469325  
## 12 styleus\_double / imperial stout -0.0581036486558732  
## 13 styledoppelbock -0.0637180602789297  
## 14 styleamerican strong ale -0.104450497361465  
## 15 styleamerican barleywine -0.106710813455614  
## 16 styleamerican stout -0.108436654785507  
## 17 stylefruit / vegetable beer -0.220055597004256  
## 18 revcount 0.00000939505824663106  
## 19 styletripel 0.00336108784121428  
## 20 styleamerican amber / red ale 0.00382728930517444  
## 21 styleamerican ipa 0.00463397739784681  
## 22 stylemarzen / oktoberfest 0.00472157396636444  
## 23 stylesaison / farmhouse ale 0.0193266699315449  
## 24 styleamerican pale wheat ale 0.0227394976215729  
## 25 styleamerican pale ale (apa) 0.027111151022055  
## 26 appearance 0.0277005574603479  
## 27 stylehefeweizen 0.036795322904399  
## 28 aroma 0.108620275260883  
## 29 palate 0.317092790008477  
## 30 taste 0.410097577647873  
## 31 (Intercept) 0.847798576786722

The smaller values of lambda produced smaller cross-validation MSE. The best lambda 0.027 produced the smallest CV-MSE of 0.007, suggesting the shrinkage is closer to the coefficients of the OLS model. The ridge model with teh best lambda has minimal bias.

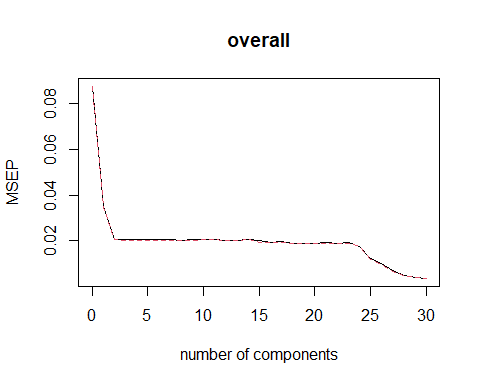
## PCR

A PCR test allows further tests of dimensionality and multicollinearity.

# library(pls)  
  
pcr\_fit <- pcr(overall ~ ., data = beer\_data, scale = T, validation = "CV")  
summary(pcr\_fit)

## Data: X dimension: 275 30   
## Y dimension: 275 1  
## Fit method: svdpc  
## Number of components considered: 30  
##   
## VALIDATION: RMSEP  
## Cross-validated using 10 random segments.  
## (Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps  
## CV 0.2955 0.1870 0.1440 0.1430 0.1429 0.1429 0.1432  
## adjCV 0.2955 0.1869 0.1429 0.1421 0.1422 0.1422 0.1425  
## 7 comps 8 comps 9 comps 10 comps 11 comps 12 comps 13 comps  
## CV 0.1433 0.1427 0.1435 0.1435 0.1435 0.1426 0.1426  
## adjCV 0.1425 0.1418 0.1427 0.1429 0.1431 0.1418 0.1418  
## 14 comps 15 comps 16 comps 17 comps 18 comps 19 comps 20 comps  
## CV 0.1428 0.1417 0.1396 0.1398 0.1378 0.1380 0.1381  
## adjCV 0.1432 0.1395 0.1383 0.1386 0.1371 0.1367 0.1371  
## 21 comps 22 comps 23 comps 24 comps 25 comps 26 comps 27 comps  
## CV 0.1385 0.1379 0.1389 0.1330 0.1101 0.09968 0.08306  
## adjCV 0.1375 0.1370 0.1378 0.1328 0.1095 0.09907 0.08258  
## 28 comps 29 comps 30 comps  
## CV 0.06952 0.06470 0.06091  
## adjCV 0.06905 0.06429 0.06047  
##   
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 comps  
## X 15.52 20.45 24.35 28.07 31.76 35.35 38.86 42.35  
## overall 60.89 78.44 79.92 79.92 79.96 79.99 80.21 80.33  
## 9 comps 10 comps 11 comps 12 comps 13 comps 14 comps 15 comps  
## X 45.84 49.32 52.79 56.25 59.69 63.13 66.56  
## overall 80.33 80.34 80.34 80.57 80.63 80.65 81.77  
## 16 comps 17 comps 18 comps 19 comps 20 comps 21 comps 22 comps  
## X 69.98 73.40 76.82 80.23 83.64 87.04 90.43  
## overall 81.95 82.06 82.35 82.61 82.62 82.67 82.70  
## 23 comps 24 comps 25 comps 26 comps 27 comps 28 comps 29 comps  
## X 93.83 96.66 98.87 99.52 99.83 99.95 99.98  
## overall 82.75 84.37 88.50 90.95 93.74 95.75 96.38  
## 30 comps  
## X 100.00  
## overall 96.81

validationplot(pcr\_fit, val.type = "MSEP")



The results of the scree plot and numeric output show the 30 component model has the best CV RMSE, showing the results are the same as the OLS and GLM model. The 30 component model is also best for interpretability and predictive accuracy.

# Conclusions

## Consumer Preference and Beverages

Several conclusions can be drawn regarding consumer beverage preferences. It turns out the drinks consumers prefer the most are the ones that they think taste the best, which we can see because ‘taste’ turns out to be the variable with the greatest positive correlation with overall score. Consumers seem to prefer hefeweizens and dislike vegetable or fruit beers, and drinks with a high ABV. However, these preferences should be taken with a grain of salt. The severe multicollinearity found for ‘taste’, ‘palate’ and ‘aroma’ indicates that these distinctions might be lost on customers, or alternatively that there’s more fuss made of these terms than is actually warranted.

## Statistical Analysis and Conclusions

Overall, the weighted GLM model is the superior model because it addresses the truncated nature of the outcome variable, overall beer rating. The weighting of coefficients also reduces the variation and produces a more stable model. As the residuals follow a Gaussian distribution, the GLM produces the same results as the OLS model, which is better for interpretability and explainability. We found that taste had the largest effect on the overall rating at 1.286 and it was the only predictor with a positive effect. The categorical predictor with the largest effect was vegetable-style beer, -0.017

## Challenges and Lessons Learned

The data set had significant number of factors, which proved both challenging and interesting to work with. Many of the models were difficult to interpret because of the sheer number of variables involved. While these many variables added significant length to the outputs, they only provided limited insights. In future studies it would make sense to first run a regression to determine the most popular category, and then run regressions within that category to find the most determinative variables. Another option would be to run preliminary models to reduce the number or effect of categorical variables such as stepwise, Ridge, or LASSO. The combination of a large number of categorical variables and several continuous variables increased the difficulty of selecting models and creating visually useful plots.

Another issue was that the number of reviews (revcount) varied significantly. This variation proved to be statistically significant in the models. That may be a useful insight on its own. However, it may make sense to weight the number of reviews per product or extract a random sample to get a similarly sized and normally distributed sample for each product so that that variation could be accounted for.

The model also demonstrated significant multicollinearity and heteroskedasticity. As was seen in the correlation matrix and BP test. So methods to account for that are key.