Beverage Review Project

Nick McCulloch

Contents

Introduction

Project Description

The purpose of this project is 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 is larger trends in consumer preferences across products so this data has been filtered to exclude products with fewer than 500 reviews, aggregated by style and filtered again to exclude any style with fewer than 5 products. The resulting dataset consists 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 are 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
```

```r
names(beer_data)
```

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

```r
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 ...
```

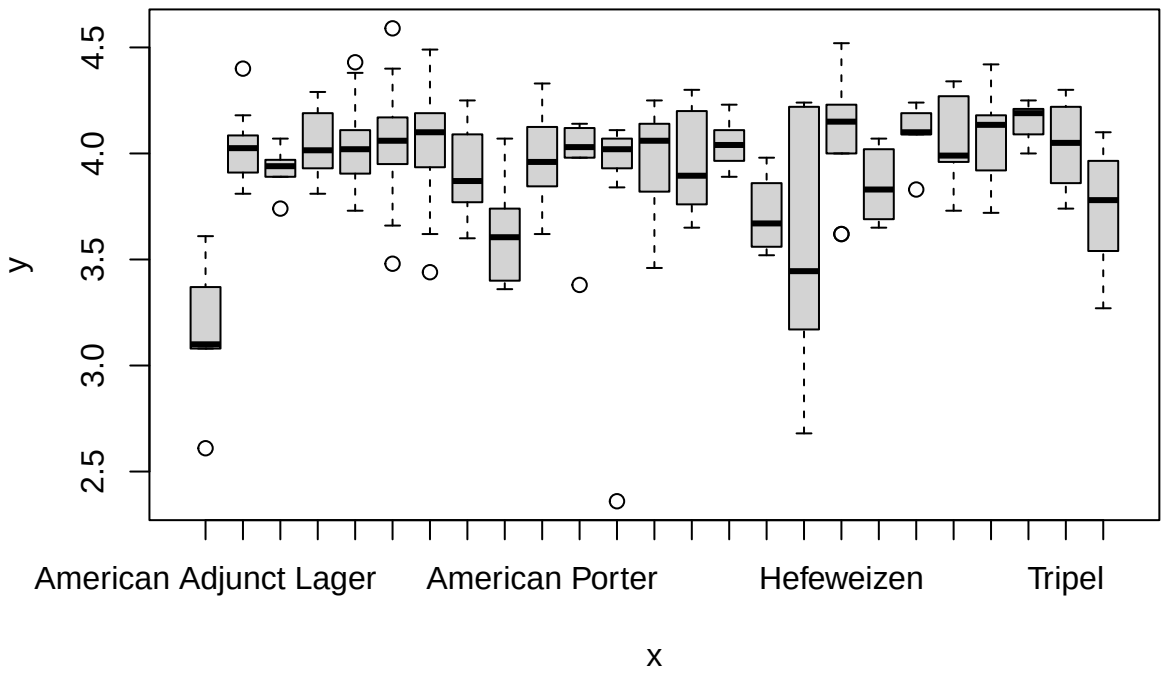
```r
data_summary <- summary(beer_data)
```

The number of reviews per product and ABV has 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 has 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. Review appearance has the highest level of positive kurtosis (6.28) followed by the overall score with 5.53.

Preliminary Tests and Transformations

Test Plot

```r
# quick check with base R plot to get sense of data.
plot(beer_data$style, beer_data$overall)
```



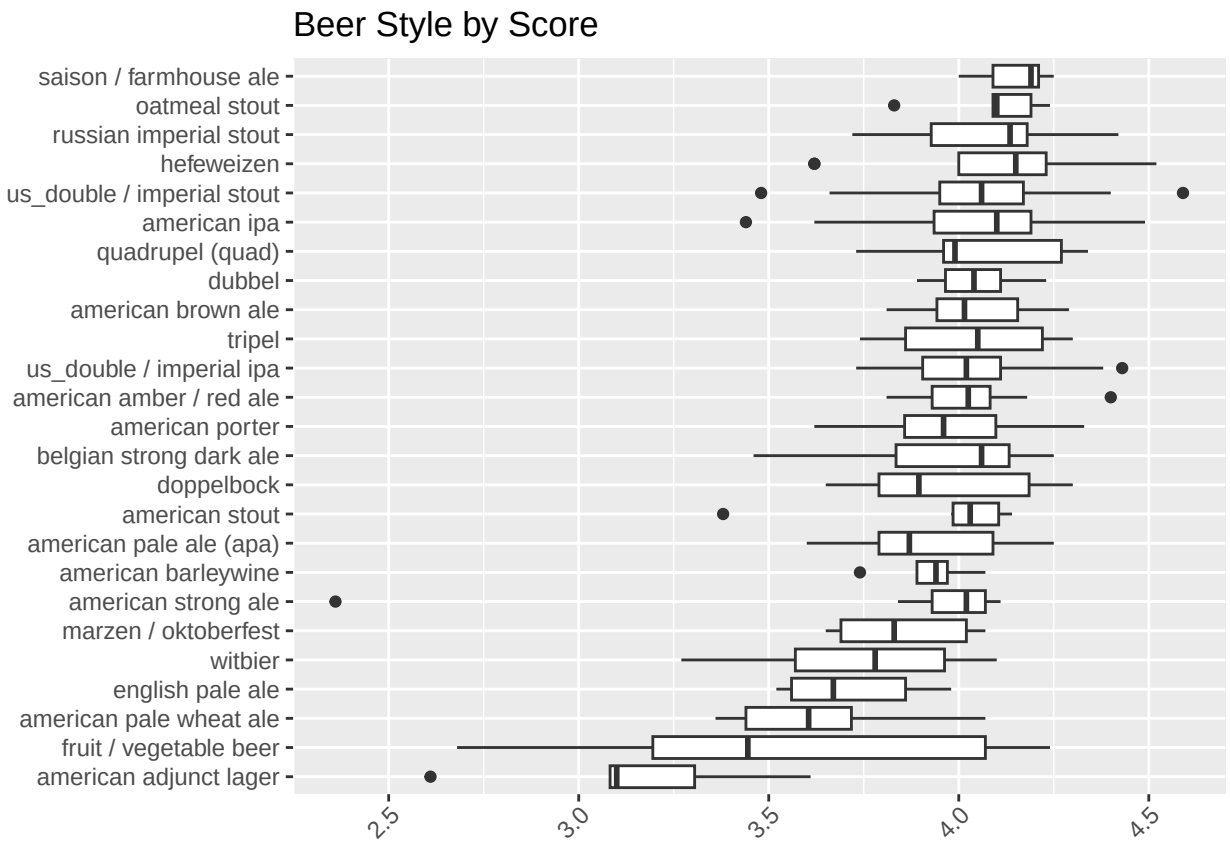
Transformations

```
# 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 =

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
```



Further Transformations

```
# creating column of average overall scores by style.
style_avg <- beer_data %>%
  group_by(style) %>%
  summarise(mean_style = mean(overall))

print(style_avg[order(style_avg$mean_style), ])
```

```
## # A tibble: 25 x 2
##    style                    mean_style
##    <fct>                         <dbl>
##  1 american adjunct lager         3.14
##  2 fruit / vegetable beer         3.53
##  3 american pale wheat ale        3.63
##  4 english pale ale               3.72
##  5 witbier                        3.74
##  6 marzen / oktoberfest           3.85
```

```
##  7 american strong ale          3.86
##  8 american barleywine          3.92
##  9 american pale ale (apa)      3.94
## 10 american stout               3.95
## # i 15 more rows
```

Basic Linear Model

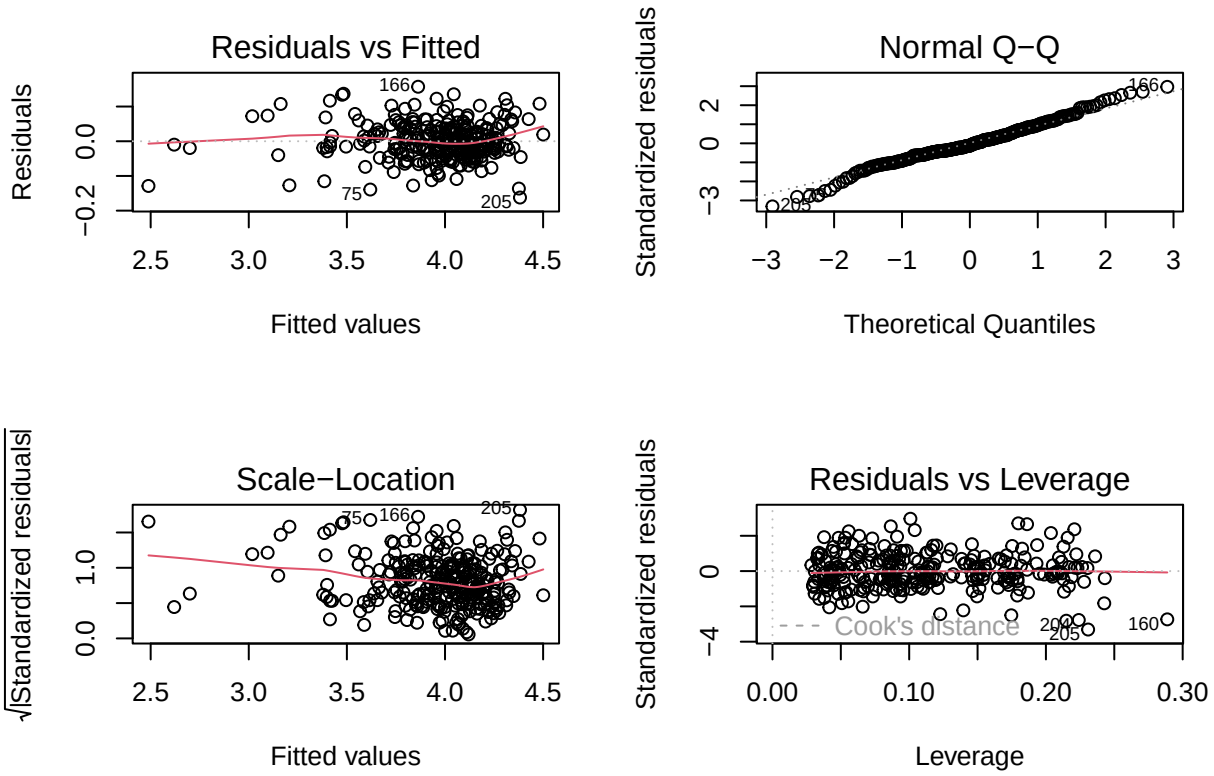
```
lm_basic <- lm(overall ~ ., data = beer_data)

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
```

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, including one that is 3.5x the mean.

```
bptest(lm_basic, data = beer_data)
```

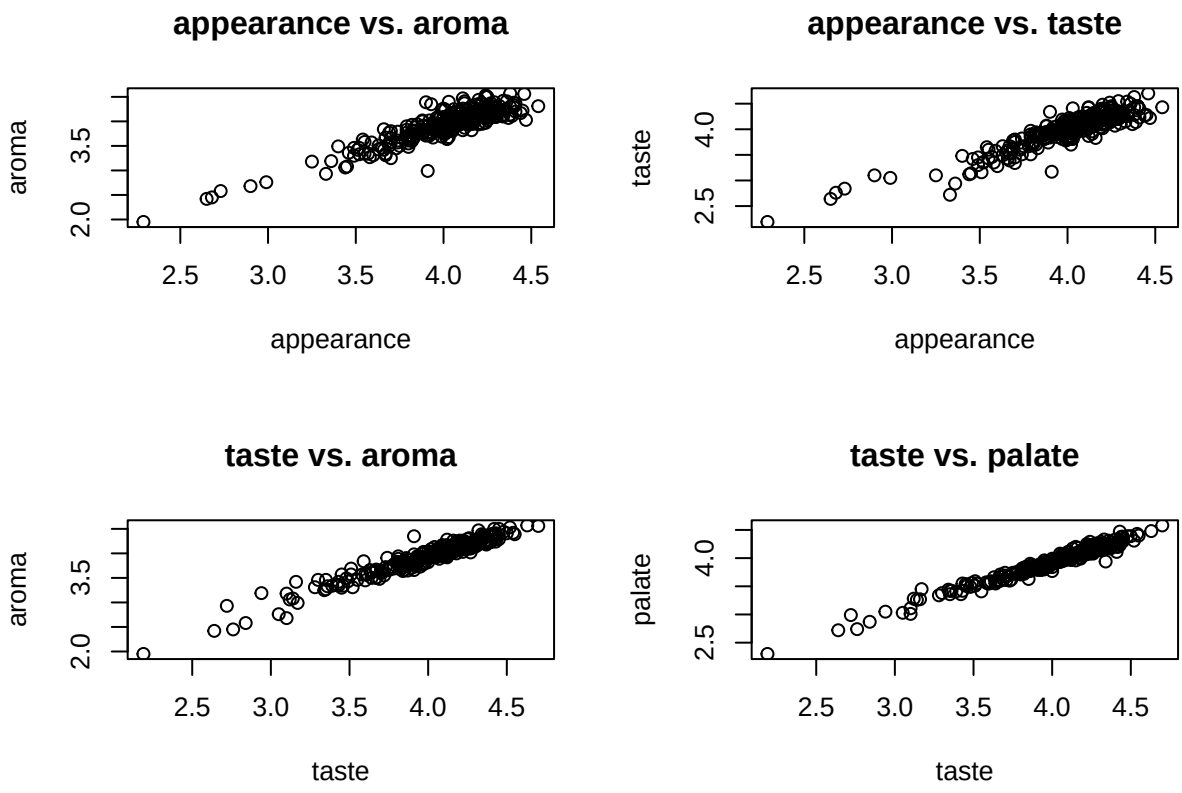
```
##
```

```
##  studentized Breusch-Pagan test
##
## data:  lm_basic
## BP = 73.433, df = 30, p-value = 1.66e-05
```

The BP test strongly suggests heteroskedasticity.

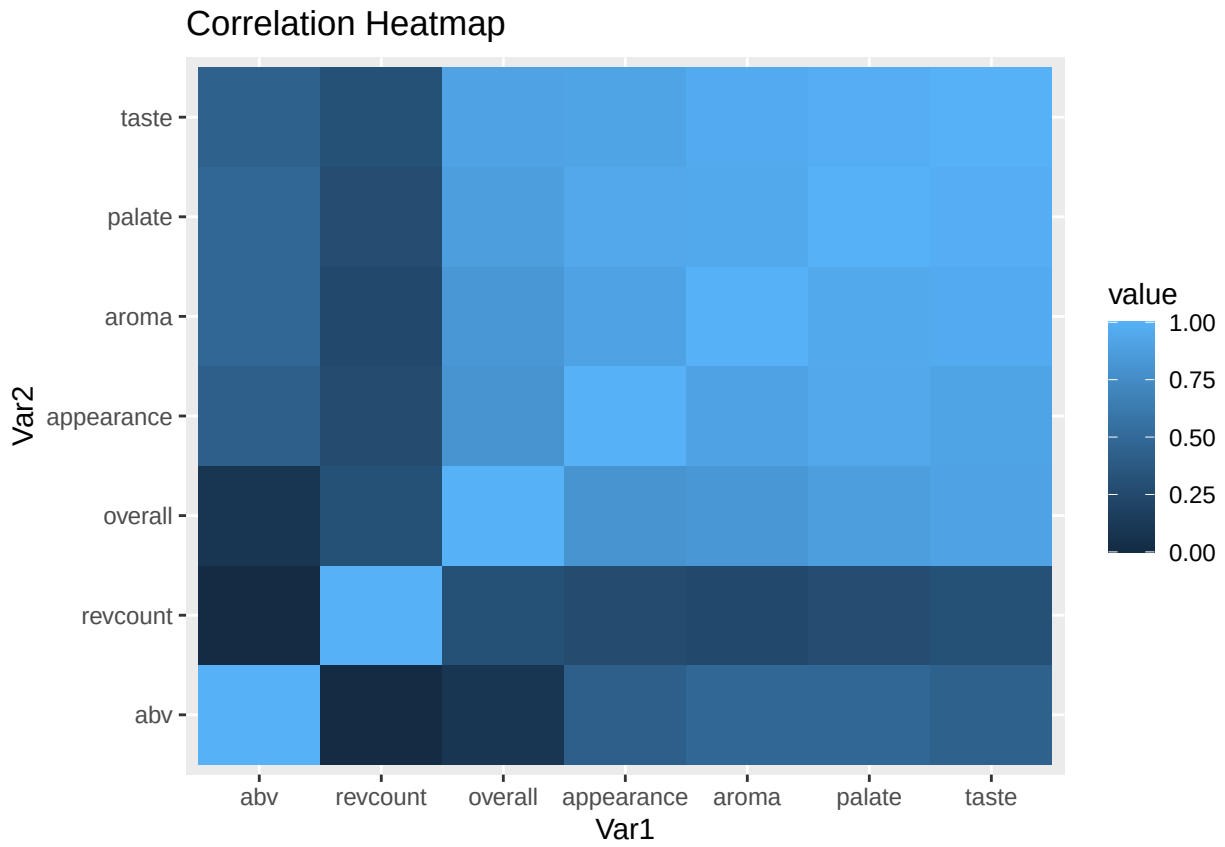
Visual Check for Multicollinearity

```
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")
```



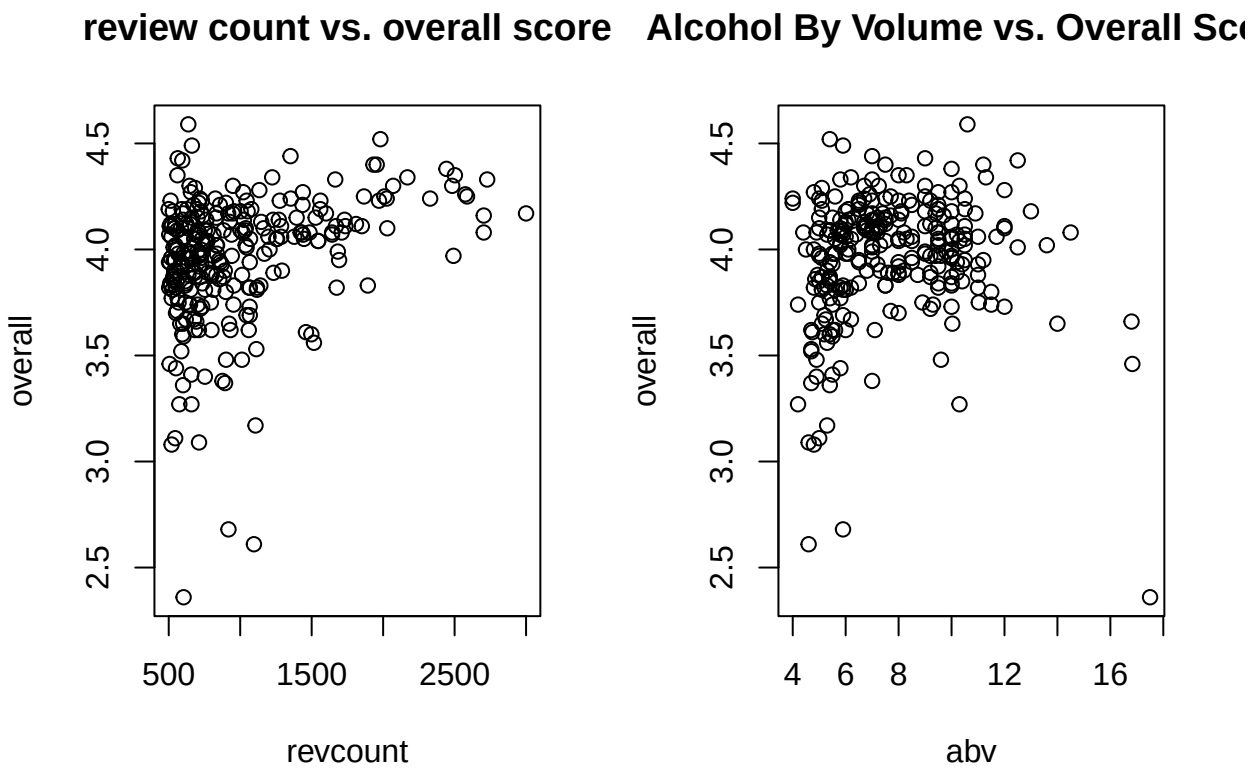

```
# excluding 'style' from the corr plot, which is the only non-numeric column
corr_data <- beer_data[-1]

# creating correlation matrix
corr_mat <- round(cor(corr_data), 2)
```

```r
# 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.

```r
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 is 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)

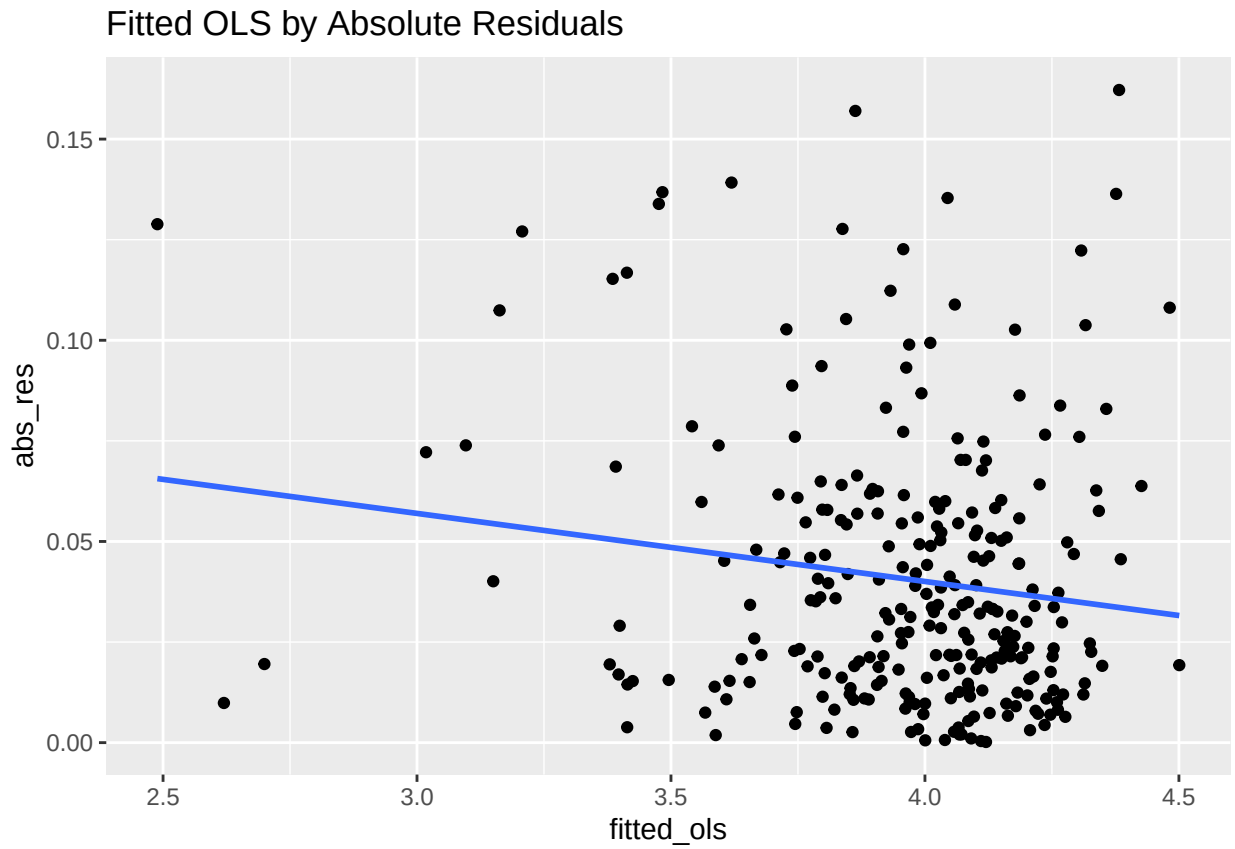
The multicollinearity and heteroskedasticity found above suggested the need to test a WLS model, and because the residuals have a Gaussian distribution, a GLM model, as well.

```
# 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
```

```r
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")
```




```r
# 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)
```

```r
# 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"
```

```r
paste("fit_wls-adjusted r-squared: ", format(sum_fit_wls$adj.r.squared, digits = 4))
```

```
## [1] "fit_wls-adjusted r-squared:  0.9548"
```

WGLM

```r
# 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$coefficients)
basic_coef <- basic_coef[!(row.names(basic_coef) %in% c("abv", "appearance", "aroma", "palate", "taste"
basic_coef <- round(basic_coef, 4)
basic_coef$coefficients <- rownames(basic_coef)
basic_coef <- basic_coef[, c(5, 1, 2, 3, 4)]

fit_coef <- data.frame(fit_sum$coefficients)
fit_coef <- fit_coef[!(row.names(fit_coef) %in% c("abv", "appearance", "aroma", "palate", "taste", "rev
fit_coef <- round(fit_coef, 4)
fit_coef$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-highe
```

```r
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

```r
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)
```

```r
# 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
```

```r
# 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 is 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: 15 172 28 43 215 229 100 168 206 152"
```

```
# 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
## 0.002710173 0.052059322
```

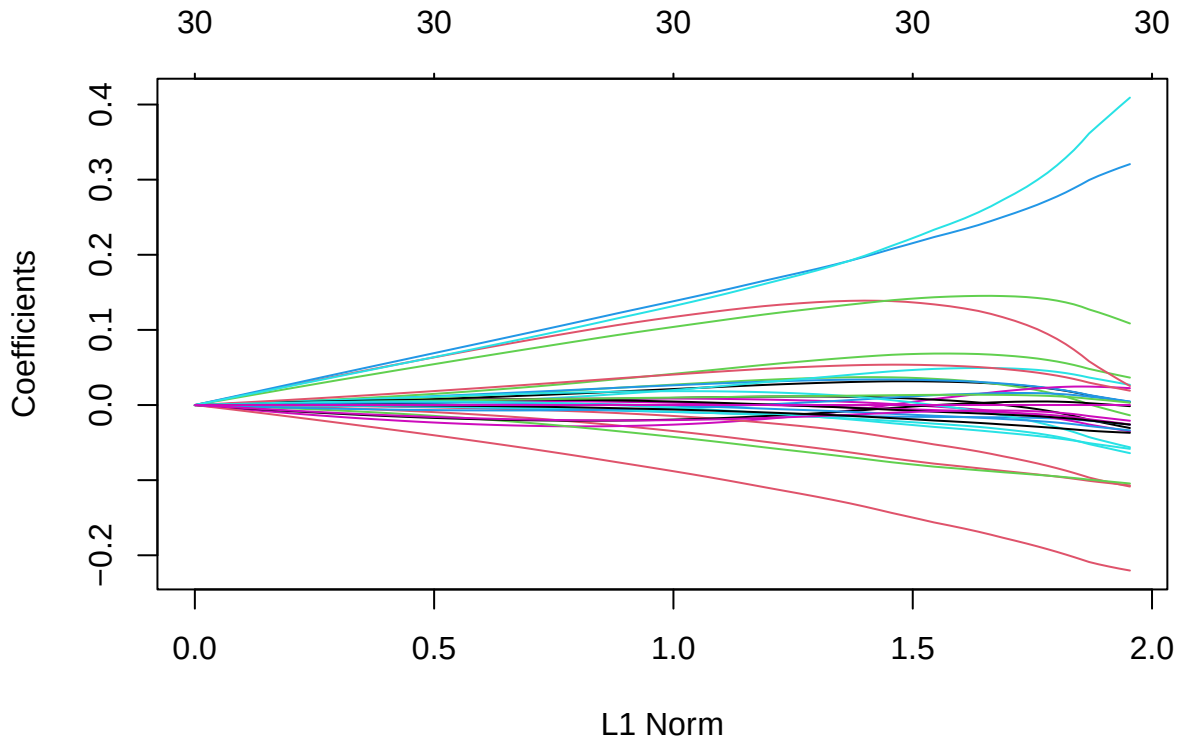
Random Split Cross-Validation Test MSE

```
# 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))
```

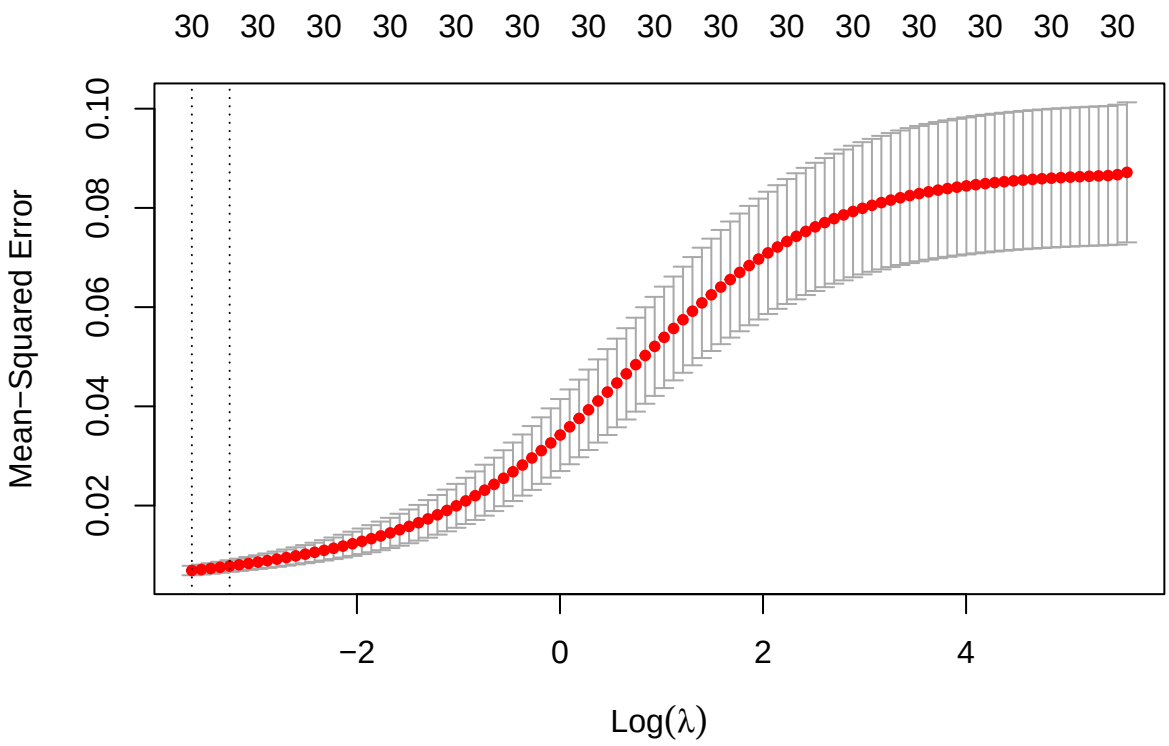
```
##  RSCV Test MSE RSCV Test RMSE
##    0.003625107    0.060208865
```

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

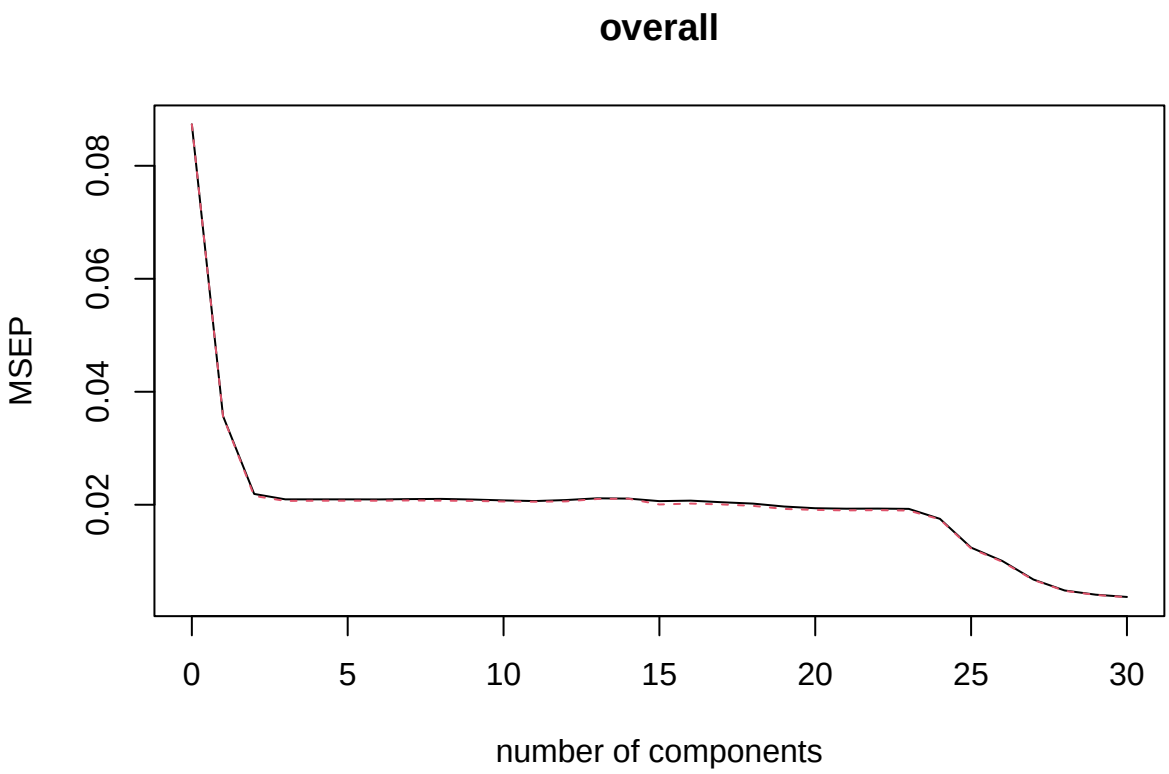
A PCR test allows further tests of dimensionality and multicollinearity.

```r
# 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   0.1890   0.1480   0.1447   0.1447   0.1447   0.1447
## adjCV       0.2955   0.1888   0.1468   0.1438   0.1439   0.1439   0.1439
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV      0.1449    0.145   0.1447    0.1441    0.1437    0.1443    0.1454
## adjCV   0.1441    0.144   0.1438    0.1434    0.1433    0.1435    0.1450
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
## CV       0.1452    0.1436    0.1440    0.1430    0.1421    0.1403    0.1392
## adjCV    0.1452    0.1416    0.1421    0.1416    0.1407    0.1388    0.1381
##        21 comps  22 comps  23 comps  24 comps  25 comps  26 comps  27 comps
## CV       0.1389     0.139    0.1388    0.1323    0.1114   0.10025   0.08221
## adjCV    0.1378     0.138    0.1377    0.1319    0.1107   0.09953   0.08174
##        28 comps  29 comps  30 comps
## CV      0.06944   0.06384   0.06064
## adjCV   0.06896   0.06342   0.06020
##
## 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 is the only predictor with a positive effect. The categorical predictor with the largest effect is 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 is that the number of reviews (revcount) varies 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 is seen in the correlation matrix and BP test. So methods to account for that are key.