Project 1 – Modeling Beer Advocate Reviews

Andrew Leppla, Huy Hoang Nguyen, Ikenna Nwaogu

I. Introduction

In this project, we will study the Beer Advocate data set [1]. Beer review data from this data set were used to address two project objectives:

1. Build predictive regression models using cross validation with metrics to compare multiple models. Provide interpretation of the regression model(s), including: hypothesis testing, interpretation of regression coefficients, and confidence intervals as well as practical vs. statistical significance.
2. Perform a secondary analysis using Time Series and address if the assumption of independent errors is valid for the final regression model.

II. Data Description

The data set from the online forum BeerAdvocate.com [1] has over 1.5 million individual beer reviews that cover 66,055 unique beers from 5,840 breweries. Most of these reviews focus on craft beers and are not representative of mass-market beers like Budweiser, Miller, Coors, etc. The reviews span about 15 years from 1996 to the end of 2011. In addition to brewery and beer names, reviews include 5 different ratings: overall, taste, appearance, aroma, and palate. This project focused on Overall Rating as the primary response. Additional data are provided on beer style, alcohol by volume (ABV), and review time.

III. Exploratory Data Analysis

Taking a first look at the data, the beer ratings are discrete with values of 1 to 5 in increments of 0.5 (see Figure III.1), 1 being the worst rating and 5 being the best. Ordinal logistic regression or a discrete choice model would be most appropriate here, but they're not within the scope of this project. With over 1.5 million reviews, the ratings can be averaged across different variables to normalize them per the Central Limit Theorem (Figure III.2). These will be called Average Reviews to distinguish them from the raw Individual Reviews.

Using Overall Rating as the response, the other 4 ratings are correlated with Overall Rating and with each other, indicating a potential issue with multicollinearity (Table III.1). This issue is exacerbated when using Average Reviews by beer name (Table III.2, Figure III.2).

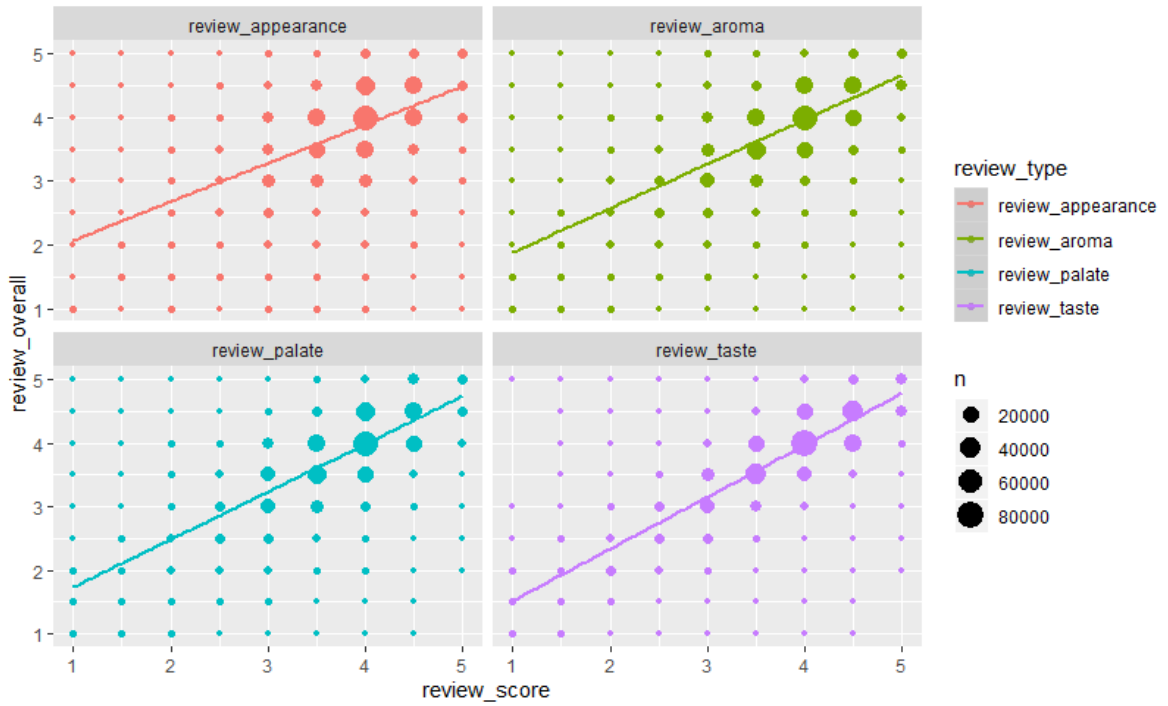

Figure III.1 – Scatterplots of Individual Overall Rating vs. other individual ratings in year 2011. Ratings data are discrete.

Table III.1 – Correlation matrix of individual review ratings data for year 2011

	review_overall	review_aroma	review_appearance	review_palate	review_taste
review_overall	1.0000000				
review_aroma	0.6872329	1.0000000			
review_appearance	0.5236376	0.4922312	1.0000000		
review_palate	0.7221078	0.5617439	0.5165652	1.0000000	
review_taste	0.8483747	0.6856092	0.4839489	0.6948401	1.0000000

Table III.2 – Correlation matrix of average review ratings by beer name for year 2011. Average reviews are more correlated.

	Overall	Aroma	Visual	Palate	Taste
Overall	1.0000000				
Aroma	0.7800110	1.0000000			
Visual	0.6272636	0.5996442	1.0000000		
Palate	0.8121640	0.6842859	0.6282888	1.0000000	
Taste	0.8984971	0.7882234	0.6015011	0.7896848	1.0000000

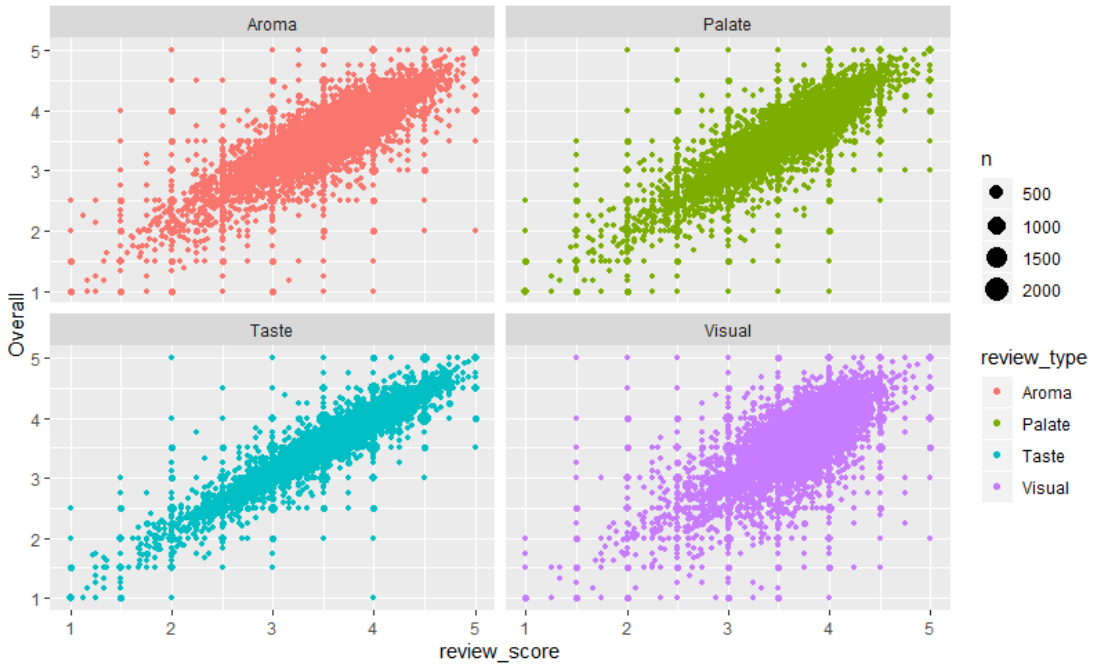

Figure III.2 – Scatterplots of Overall Rating vs. other ratings averaged by beer name in 2011. Average Ratings data are more normalized and more correlated.

ABV data are not normal and have a long right-tailed distribution (Figure III.3). It also has a relatively weak correlation with Average Overall Rating and a few non-linear outliers (Figure III.4). Transformation was unsuccessful, so this will be addressed by checking residuals for normality as well as leverage and influence (Cook's D) for any modeling with ABV as a factor.

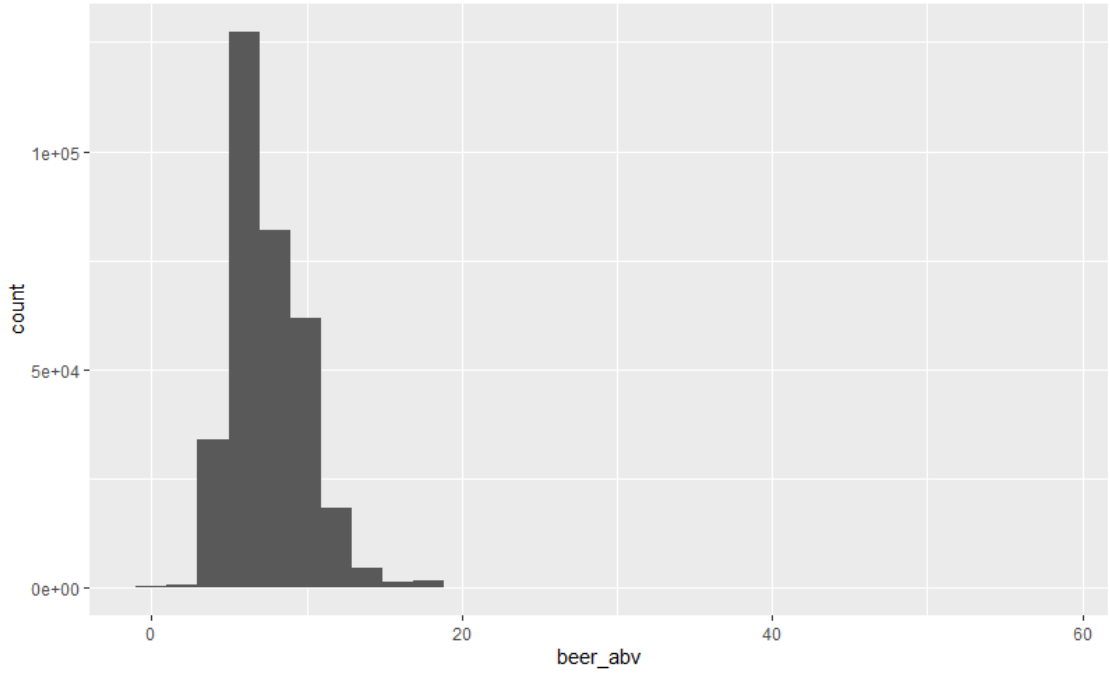

Figure III.3 – Right-skewed ABV data.

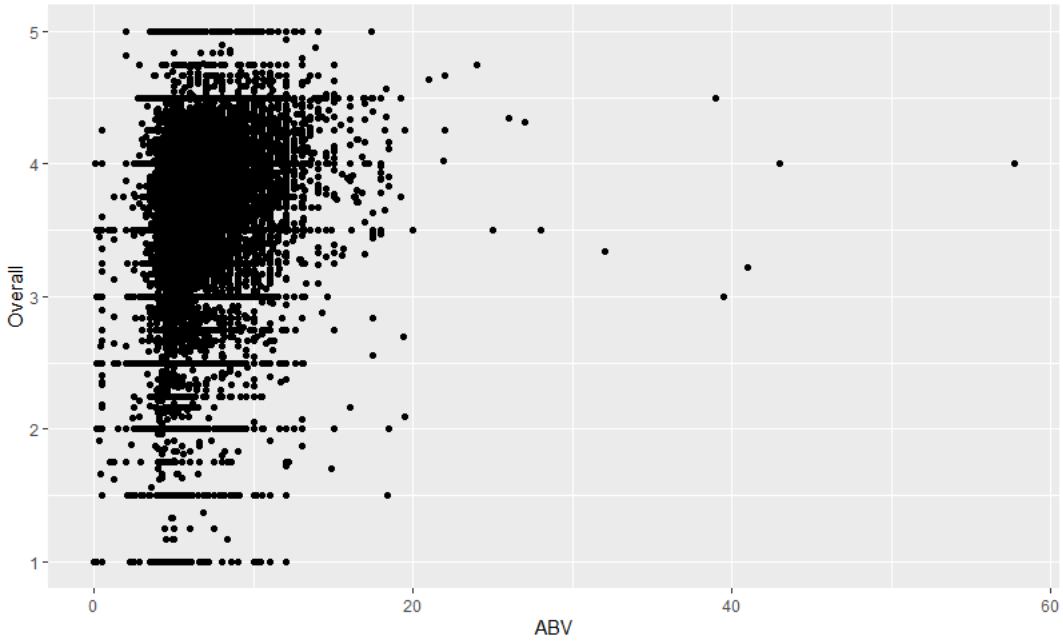

Figure III.4 – Overall Ratings vs. ABV averaged by beer name in 2011. Some non-linear outliers and weakly correlated.

The most reviewed individual beer was Dogfish Head's "90 Minute IPA". The Overall reviews of Dogfish Head's "90 Minute IPA" were plotted vs. time to explore the time series data. Per Figure III.5, the review data are not spaced equally over time (a key assumption of time series analysis). This is expected given that beers aren't being reviewed every second of every day in the primarily U.S.-based market.

Unequal data spacing over time was addressed by averaging the reviews by Month to ensure at least one review for any given time spacing. This also normalized the Overall Rating data. See Figure III.6. Unequal spacing could also be addressed by assigning each review an index value rather than a date and time value, but there would still be an issue with discrete individual review data.

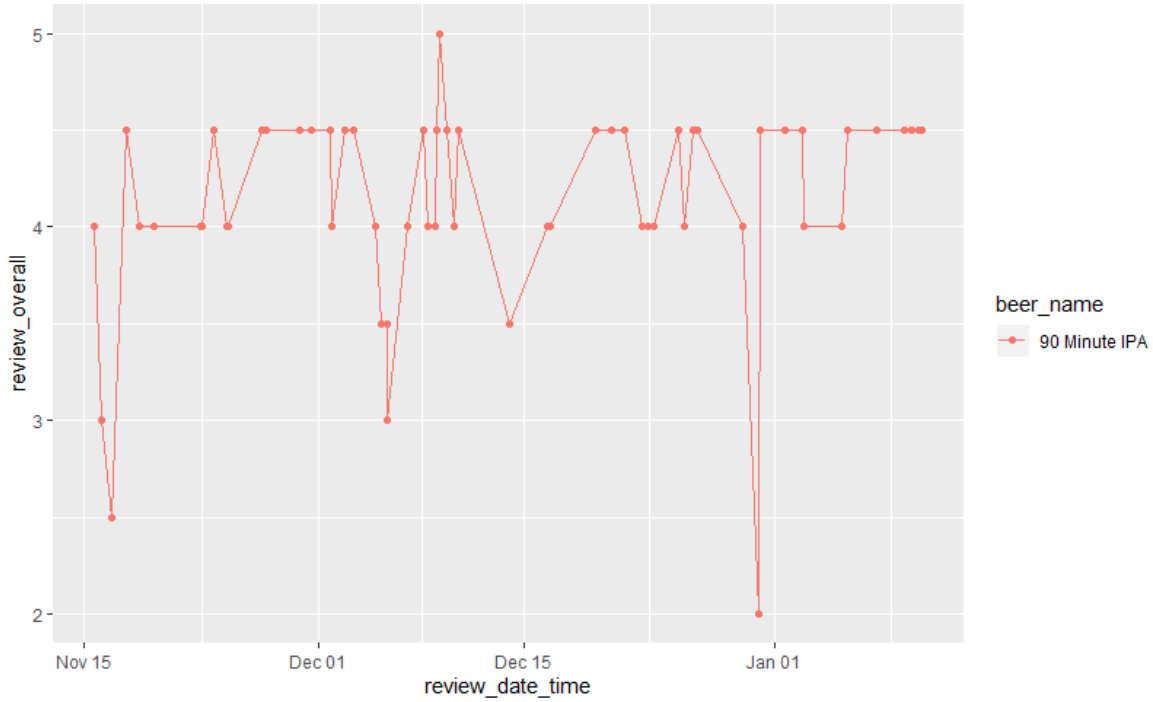

Figure III.5 – Individual Overall Reviews for beer "90 Minute IPA" in Winter 2011. Data are discrete and not equally spaced across time.

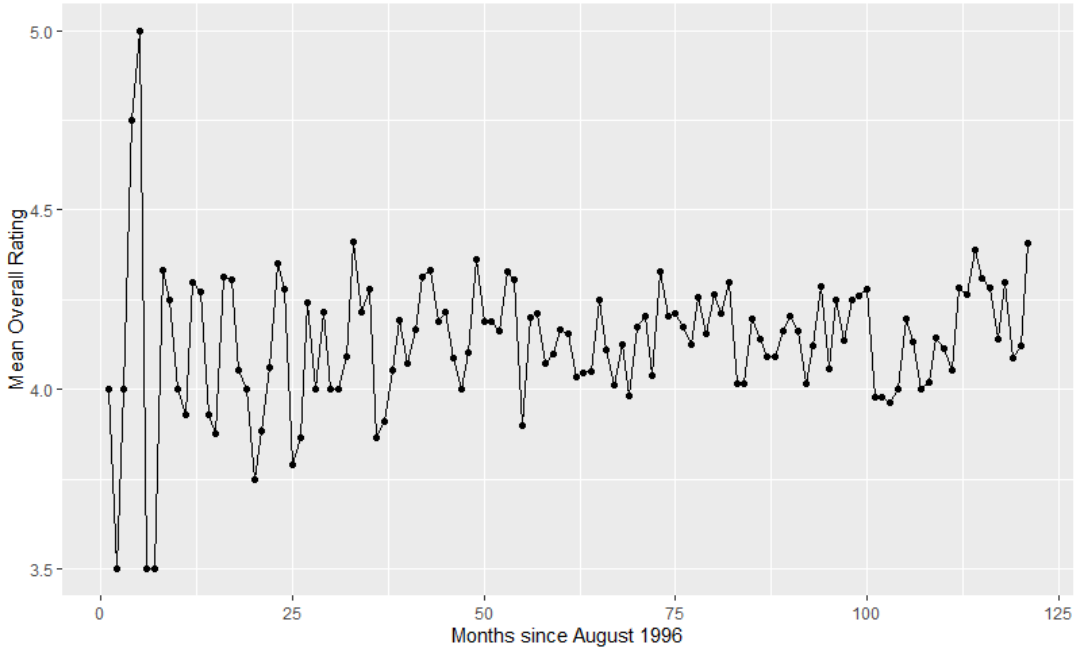

Figure III.6 – Average Overall Reviews for beer "90 Minute IPA" by Month. Data are equally spaced across time and more normalized.

IV. Objective 1

Problem Statement

Multiple Regression Analysis (*cf.* [2-10]) is the study of how a dependent variable is related to two or more independent variables. In the next steps, we will build predictive regression models using cross validation with metrics to compare multiple models for the Beer Advocate dataset. We will discuss and provide interpretation of the chosen regression models including hypothesis testing, interpretation of regression coefficients and confidence intervals as well as practical vs. statistical significance.

Model Selection

In our Multiple Regression Analysis, we chose to look at the review data averaged by each unique beer (brewery and beer name). We aggregated and counted the number of reviews each beer received and used that as an additional predictor to see if it would help with a better model. Before we started our model analysis, we removed some missing data on ABV and then examined the correlations for all predictors including count using the scatter plots in Figure IV.1.

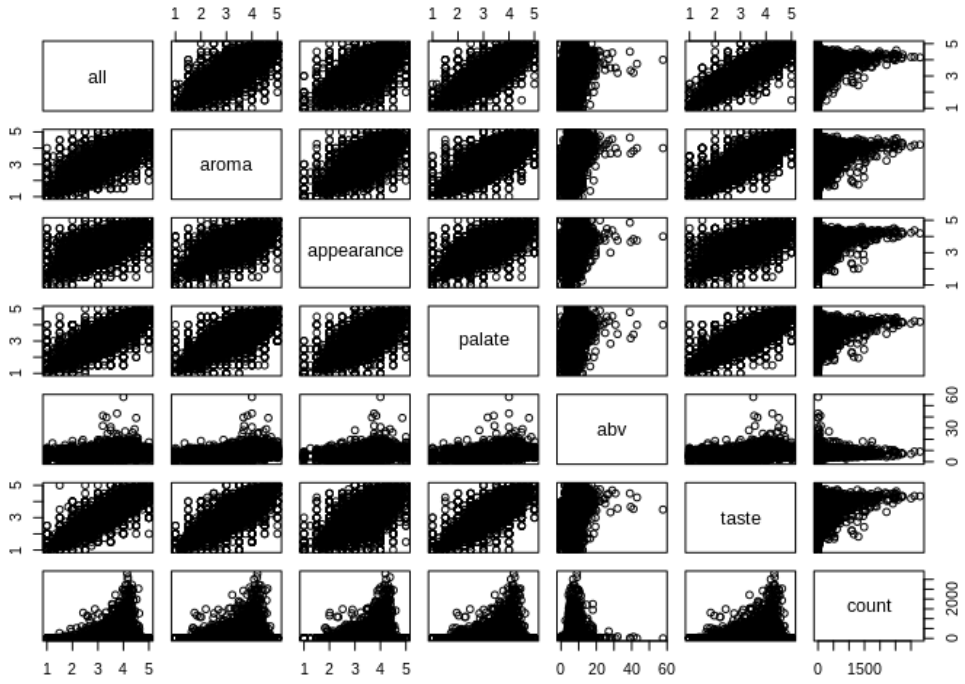

Figure IV.1 – Regression Model Residual plot with the inclusion of Count. all = overall.

The plots in Figure IV.1 indicate that count and ABV are not highly correlated to the other features, but we are going to keep ABV because we know that ABV is generally an important indicator in alcoholic beverages. The plots in Figure IV.2 and Figure IV.3 also show that the partial residuals for count and log(count) have nonconstant variance which is problematic.

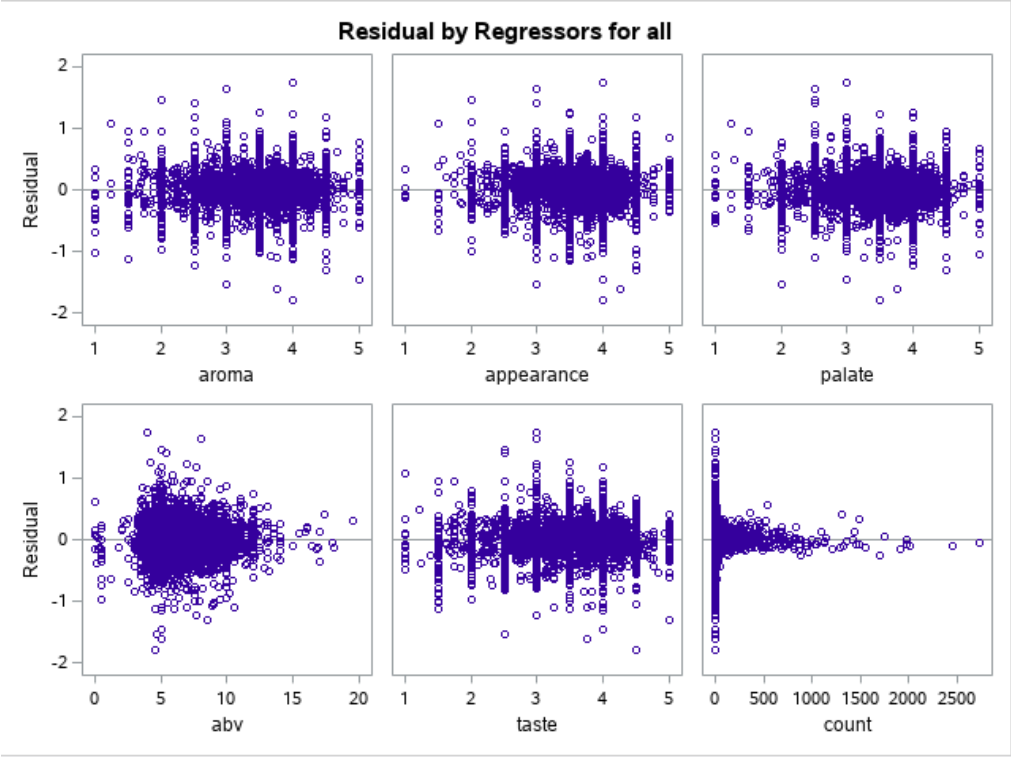

Figure IV.2 – Regression Model Partial Residual plots with the inclusion of count

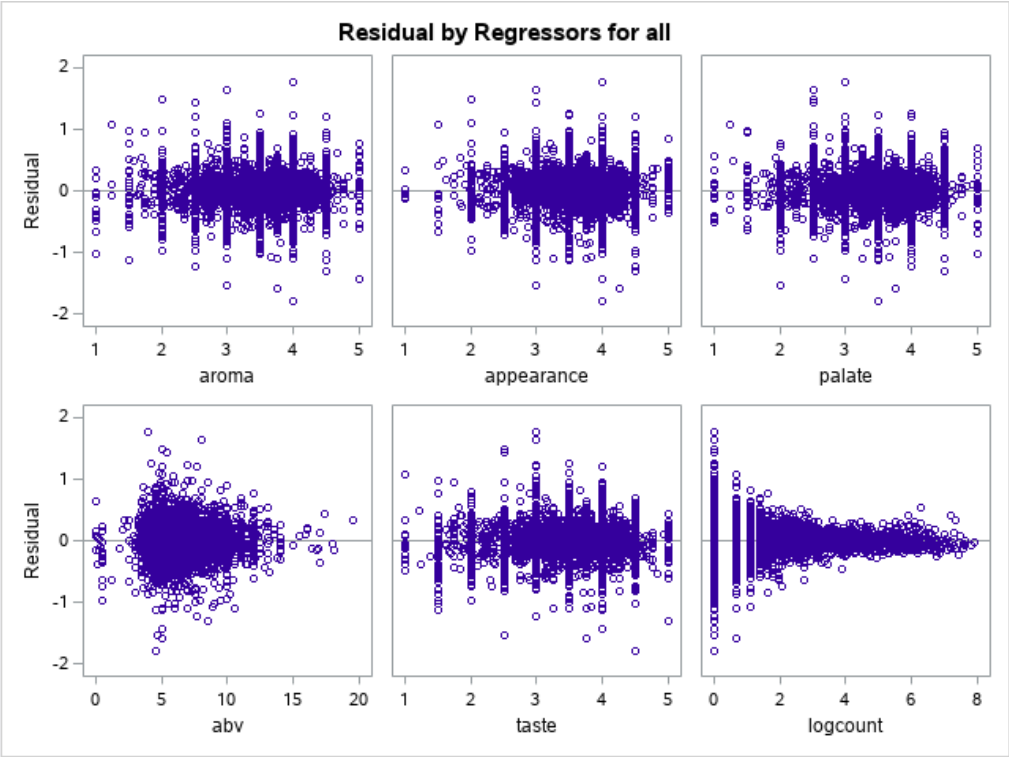

Figure IV.3 – Regression Model Partial Residual plots with the inclusion of log(count)

The data also shows a heavily skewed count which cannot be fixed using transformation as you can see in the histogram plot in Figure IV.4. Therefore, count will be excluded from modeling.

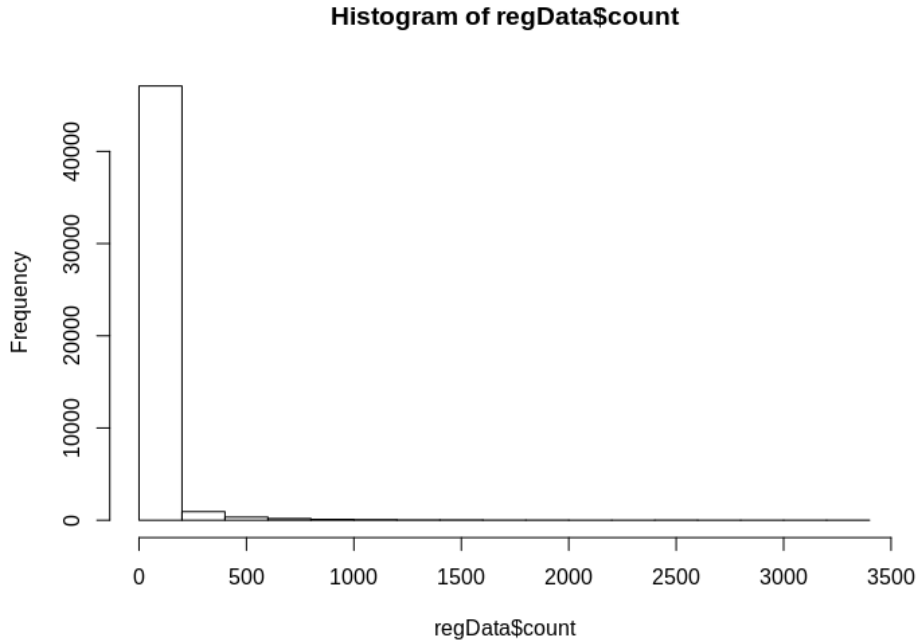

Figure IV.4 – Histogram of count (number of reviews per beer)

We used forward, stepwise, backward and a mixture of both forward and backward selection types for our model selection analysis. We used the mixture of both forward and backward AIC selection methods to check the importance of the selected variables as well as the interactions between them. We first ran variable selection for the main 5 predictors and plotted the importance shown in Figure IV.6.

Assumptions

The errors of the regression model must be normally distributed, have constant variance, and be statistically independent from each other. The QQ plot shows some deviations from normality at the tails but the histogram looks reasonably normal, and we have enough data to invoke the Central Limit Theorem for robustness to non-normality.

Constant variance looks reasonable from the Residual vs. Predicted Value plots except at the low & high rating limits of 1 and 5, seen as diagonal boundaries. Most of the average ratings fall between 2 and 4 and have constant variance.

We will assume that the data are independent. If a beer is rated without looking at other ratings first then this is likely valid. However, if the reviewer can see any information on previous ratings beforehand then this is likely not valid. Also, we have seen in the Exploratory section that the predictors are not all independent from each other. We will proceed with caution.

By influential point analysis, there are no significant outliers with high leverage in the residual plots. We also analyzed Cook's D values which is a good measure of the influence of an observation. It is proportional to the sum of the squared differences between predictions made with all observations in

the analysis and predictions made leaving out the observation in question. A common rule of thumb is that if the Cook's D value is lower than 0.03 then it has very little influence. Refer to Figure IV.5 for more details.

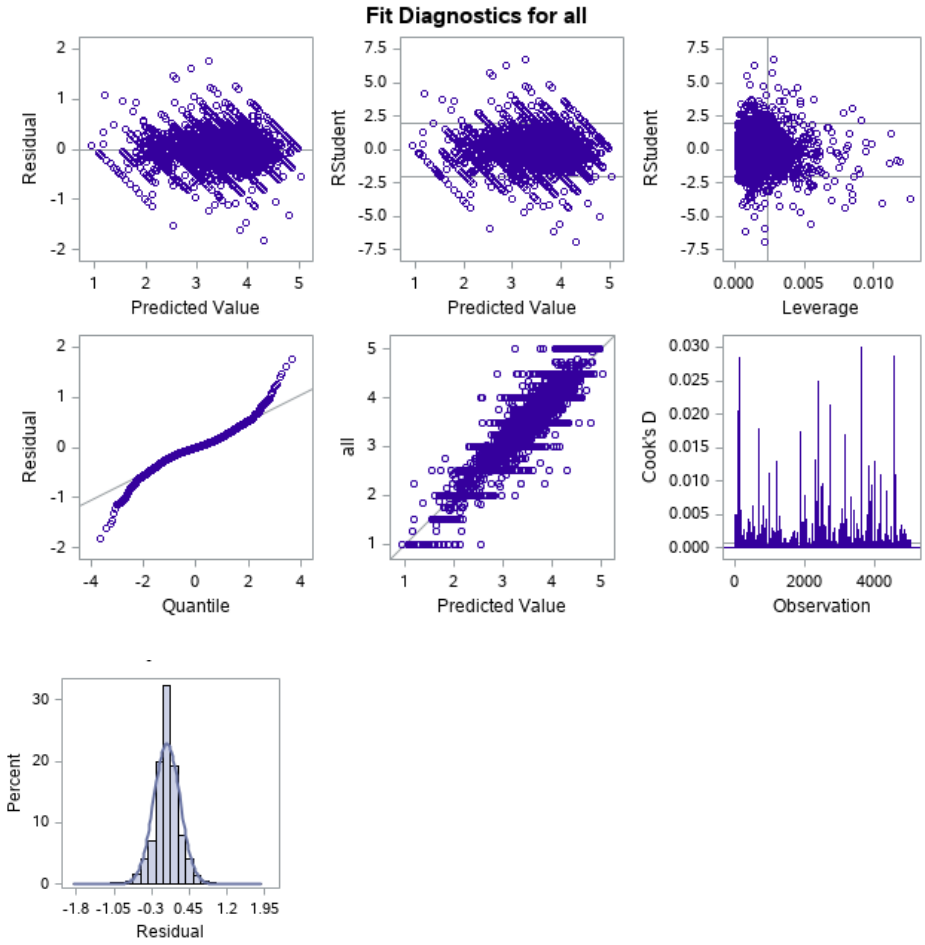

Figure IV.5 –Fit diagnostics of final model

Model Analysis

We added 3 interaction terms using combinations of the top 3 significant variables and plotted their importance using forward and stepwise methods which are shown in Figures IV.7 and IV.8, respectively. We acknowledge that palate and taste might be too highly correlated to model their interaction without significantly inflating the variance, but we added it to our analysis to see what the comparisons would yield. We then used the olsrr package in R (a set of tools for improved output from linear regression models) to compare all the possible model combinations using the 8 features. The package ran 255 different combinations and we plotted the 8 models with the best AIC, SBIC, SBC, R-Square, Adj. R-square and CP values. The plots are shown in Figures IV.11, IV.12 and IV.13 respectively. The output below shows the predictors that performed better.

```
> subset
                          Best Subsets Regression
--------------------------------------------------------------------------------
Model Index    Predictors
--------------------------------------------------------------------------------
     1         taste
     2         palate taste
     3         palate taste palate:abv
     4         aroma palate abv taste
     5         aroma palate abv taste palate:taste
     6         aroma appearance palate abv taste palate:taste
     7         aroma appearance palate abv taste abv:taste palate:taste
     8         aroma appearance palate abv taste abv:taste palate:abv palate:taste
--------------------------------------------------------------------------------

                                                Subsets Regression Summary
--------------------------------------------------------------------------------------------------------------------------------------------
                      Adj.        Pred
Model    R-Square     R-Square    R-Square      C(p)          AIC           SBIC          SBC         MSEP      FPE       HSP       APC
--------------------------------------------------------------------------------------------------------------------------------------------
  1        0.7731      0.7731      0.7731    9713.5241    18399.8139    -120132.0257    18426.2013    0.0853    0.0853    0.0000    0.2269
  2        0.7984      0.7984      0.7984    3179.6106    12621.0687    -125908.2601    12656.2520    0.0758    0.0758    0.0000    0.2016
  3        0.8068      0.8068      0.8067    1027.8372    10557.6855    -127968.6236    10601.6646    0.0727    0.0727    0.0000    0.1933
  4        0.8097      0.8097      0.8096     281.7358     9821.3375    -128701.8244     9874.1124    0.0716    0.0716    0.0000    0.1904
  5        0.8103      0.8103      0.8102     125.4232     9665.6469    -128854.3933     9727.2175    0.0714    0.0714    0.0000    0.1898
  6        0.8107      0.8107      0.8106      12.6149     9552.9701    -128963.9330     9623.3366    0.0712    0.0712    0.0000    0.1893
  7        0.8108      0.8107      0.8106      10.6623     9551.0172    -128962.7817     9630.1795    0.0712    0.0712    0.0000    0.1893
  8        0.8108      0.8107      0.8106       9.0000     9549.3544    -128961.3396     9637.3125    0.0712    0.0712    0.0000    0.1893
--------------------------------------------------------------------------------------------------------------------------------------------
AIC: Akaike Information Criteria
 SBIC: Sawa's Bayesian Information Criteria
 SBC: Schwarz Bayesian Criteria
 MSEP: Estimated error of prediction, assuming multivariate normality
 FPE: Final Prediction Error
 HSP: Hocking's Sp
 APC: Amemiya Prediction Criteria
```

We also compared our selection method with BIC, ASE, adjusted R-square, and RSS methods using 75% train and 25% test split; the plots can be found in Figure IV.9 and Figure IV.10.

We used AIC and RMSE for our final model selection. We ran backward, forward and stepwise methods with AIC and also used K-fold for cross validation. Backward and forward methods resulted in 7 predictor variables while the stepwise method resulted in 8 predictor variables.

Stepwise Summary Results

```
> step.model$results
  nvmax      RMSE  Rsquared       MAE      RMSESD  RsquaredSD       MAESD
1     1 0.2921048 0.7730939 0.2008814 0.005115534 0.007010831 0.003807852
2     2 0.2753188 0.7984693 0.1945492 0.005256912 0.006296195 0.003658910
3     3 0.2696127 0.8067437 0.1879098 0.005102957 0.006280105 0.003545351
4     4 0.2863562 0.7804271 0.1999922 0.024507427 0.039257620 0.017159608
5     5 0.2671254 0.8102966 0.1857818 0.004949889 0.006432667 0.003465688
6     6 0.2670379 0.8104191 0.1857957 0.004854571 0.006373339 0.003377996
7     7 0.2668888 0.8106325 0.1856207 0.004922345 0.006375342 0.003553200
8     8 0.2668395 0.8107042 0.1855099 0.004852617 0.006277878 0.003455379
```

```
> summary(step.model$finalModel)
Subset selection object
8 Variables  (and intercept)
             Forced in Forced out
aroma            FALSE      FALSE
appearance       FALSE      FALSE
palate           FALSE      FALSE
abv              FALSE      FALSE
taste            FALSE      FALSE
abv:taste        FALSE      FALSE
palate:abv       FALSE      FALSE
palate:taste     FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: 'sequential replacement'
         aroma appearance palate abv taste abv:taste palate:abv palate:taste
1  ( 1 ) " "    " "        " "     " " "*"   " "       " "        " "
2  ( 1 ) " "    " "        "*"     " " "*"   " "       " "        " "
3  ( 1 ) " "    " "        "*"     " " "*"   " "       "*"        " "
4  ( 1 ) "*"    " "        "*"     "*" "*"   " "       " "        " "
5  ( 1 ) "*"    " "        "*"     "*" "*"   " "       " "        "*"
6  ( 1 ) "*"    "*"        "*"     "*" "*"   " "       " "        "*"
7  ( 1 ) "*"    "*"        "*"     "*" "*"   "*"       " "        "*"
8  ( 1 ) "*"    "*"        "*"     "*" "*"   "*"       "*"        "*"
```

```
> step.model$bestTune
  nvmax
8     8
```

Forward Summary Results

```
> step.model$results
  nvmax      RMSE  Rsquared       MAE      RMSESD RsquaredSD       MAESD
1     1 0.2921144 0.7729922 0.2008848 0.004741311 0.01097788 0.002735951
2     2 0.2753192 0.7983134 0.1945514 0.005168726 0.01038199 0.002084326
3     3 0.2695968 0.8065948 0.1879251 0.005512024 0.01065781 0.002642738
4     4 0.2675798 0.8094534 0.1865379 0.005824563 0.01106623 0.003016463
5     5 0.2672850 0.8098747 0.1860904 0.005780453 0.01094782 0.003011122
6     6 0.2669683 0.8103232 0.1857109 0.005734642 0.01089130 0.003036195
7     7 0.2667917 0.8105720 0.1855002 0.005753958 0.01094741 0.002999287
8     8 0.2668081 0.8105506 0.1854973 0.005760970 0.01091605 0.002994888
> step.model$bestTune
  nvmax
7     7
> summary(step.model$finalModel)
Subset selection object
8 Variables  (and intercept)
             Forced in Forced out
aroma            FALSE      FALSE
appearance       FALSE      FALSE
palate           FALSE      FALSE
abv              FALSE      FALSE
taste            FALSE      FALSE
abv:taste        FALSE      FALSE
palate:abv       FALSE      FALSE
palate:taste     FALSE      FALSE
1 subsets of each size up to 7
Selection Algorithm: forward
          aroma appearance palate abv taste abv:taste palate:abv palate:taste
1  ( 1 ) " "    " "        " "     " " "*"   " "       " "        " "
2  ( 1 ) " "    " "        "*"     " " "*"   " "       " "        " "
3  ( 1 ) " "    " "        "*"     " " "*"   " "       "*"        " "
4  ( 1 ) "*"    " "        "*"     " " "*"   " "       "*"        " "
5  ( 1 ) "*"    "*"        "*"     " " "*"   " "       "*"        " "
6  ( 1 ) "*"    "*"        "*"     " " "*"   " "       "*"        "*"
7  ( 1 ) "*"    "*"        "*"     "*" "*"   " "       "*"        "*"
> 
```

Backward Summary Results

```
> step.model$results
  nvmax      RMSE  Rsquared       MAE      RMSESD  RsquaredSD       MAESD
1     1 0.2920852 0.7732687 0.2008920 0.006723137 0.008062563 0.003984013
2     2 0.2753074 0.7985391 0.1945530 0.006049092 0.007061168 0.003531941
3     3 0.2695608 0.8068757 0.1878126 0.006316000 0.007019814 0.003782311
4     4 0.2675303 0.8097808 0.1864644 0.006298699 0.006875782 0.003762341
5     5 0.2671125 0.8103851 0.1857900 0.006172843 0.006688029 0.003638262
6     6 0.2668040 0.8108205 0.1855248 0.006220268 0.006717741 0.003641113
7     7 0.2668031 0.8108211 0.1855054 0.006231888 0.006727103 0.003648099
8     8 0.2668128 0.8108083 0.1855105 0.006239106 0.006731598 0.003651043
>
> step.model$bestTune
  nvmax
7     7
>
> summary(step.model$finalModel)
Subset selection object
8 Variables  (and intercept)
             Forced in Forced out
aroma            FALSE      FALSE
appearance       FALSE      FALSE
palate           FALSE      FALSE
abv              FALSE      FALSE
taste            FALSE      FALSE
abv:taste        FALSE      FALSE
palate:abv       FALSE      FALSE
palate:taste     FALSE      FALSE
1 subsets of each size up to 7
Selection Algorithm: backward
          aroma appearance palate abv taste abv:taste palate:abv palate:taste
1  ( 1 ) " "    " "        " "    " " "*"   " "       " "        " "
2  ( 1 ) " "    " "        "*"    " " "*"   " "       " "        " "
3  ( 1 ) " "    " "        "*"    "*" "*"   " "       " "        " "
4  ( 1 ) "*"    " "        "*"    "*" "*"   " "       " "        " "
5  ( 1 ) "*"    " "        "*"    "*" "*"   " "       " "        "*"
6  ( 1 ) "*"    "*"        "*"    "*" "*"   " "       " "        "*"
7  ( 1 ) "*"    "*"        "*"    "*" "*"   "*"       " "        "*"
```

We decided to go with all 8 predictor variables for our model because it had the least RMSE. The interaction variable of palate and taste appears to have little influence on the model variance because the RMSE of models without the interaction predictor variables are close in value. The equation below shows all the coefficients of our final model:

y = -0.034 + 0.09*aroma + 0.038*appearance + 0.402*palate - 0.035*abv + 0.653* taste - 0.004*palate*abv + 0.005*abv*taste - 0.028*palate*taste

Parameter Interpretation

Because the predictors are partially correlated, it is difficult to provide a clear interpretation of the model coefficients. Principal component analysis could help but is outside the scope of this project. As an exercise, coefficients will be interpreted **as if they were independent**. This would only be approximately true for ABV which is the least correlated predictor vs. the others in the model (r=0.2)

-0.034 (intercept) = The mean overall beer rating is -0.034 (approximately 0) when all predictors are 0. There is no practical interpretation of this since the ratings are all on a scale of 1 to 5 and no beer is truly 0% ABV due to the limitations of alcohol and water separation processes.

0.09*aroma = When the mean aroma rating for a beer is increased by 1 while holding the other predictors constant, the mean overall rating increases by 0.09

0.038*appearance = When the mean appearance rating for a beer is increased by 1 while holding the other predictors constant, the mean overall rating increases by 0.038

0.402*palate = When the mean palate rating is increased by 1 while ABV is 0% and the other predictors are held constant, the mean overall rating increases by 0.402. Again, ABV is never truly 0% but it can be close to 0%.

-0.035*abv = When the mean ABV is increased by 5% while palate and taste are 0 and the other predictors are held constant, the mean overall rating decreases by (0.035*5) = 0.175. Again, ratings are 1 to 5 so this is not really meaningful.

0.653*taste = When the mean appearance rating for a beer is increased by 1 while holding the other predictors constant, the mean overall rating increases by 0.653

-0.004*palate*abv = There is an interaction between palate and abv. When abv increases by 5%, the term 0.402*palate decreases to (0.402-0.004*5) = 0.382*palate while holding taste constant. When mean palate rating increases by 1, the term -0.035*abv decreases to (-0.035-0.004) = -0.039*abv while holding taste constant.

0.005*abv*taste = There is an interaction between abv and taste. When abv increases by 5%, the coefficient 0.653*taste increases to (0.653+0.005*5) = 0.678*taste while holding palate constant. When mean taste rating increases by 1, the term -0.035*abv increases to (-0.035+0.005) = -0.030*abv while holding palate constant.

-0.028*palate*taste = There is an interaction between palate and taste. When mean palate rating increases by 1, the term 0.653*taste decreases to (0.653-0.028) = 0.625*taste while holding abv constant. When mean taste rating increases by 1, the term 0.402*palate decreases to (0.402-0.028) = 0.374*palate while holding abv constant.

Confidence Intervals for Model Coefficients

We also included the confidence interval of all the coefficients below. The confidence intervals show that all predictors (with or without interaction terms) do not contain zero at 95% level (except *palate:abv*). This indicates that we can reject the null hypothesis and these estimates are significantly different from zero (except *palate:abv*).

```
> confint(model1)
                    2.5 %        97.5 %
(Intercept)  -0.092447542  2.363016e-02
aroma         0.082764166  9.776666e-02
appearance    0.030950460  4.462110e-02
palate        0.375529645  4.294184e-01
abv          -0.042847493 -2.699956e-02
taste         0.626279596  6.788031e-01
palate:abv   -0.007261209  8.679768e-05
abv:taste     0.001282319  8.066281e-03
palate:taste -0.033005521 -2.360950e-02
```

Conclusion

We ran different model selections in order to come up with the best model for predicting the mean overall reviews for a given beer from the reviews data. We included all 4 review (aroma, appearance, taste and palate) predictors as well as ABV and 3 interaction terms with different combinations of the top 3 significant predictors which were taste, palate and ABV.

Not surprisingly, taste was the most important factor when predicting the overall beer rating. If a beer tastes good, people like it and vice versa. Palate (mouthfeel) was also important and was positively correlated with, and negatively interacted with, taste. If a beer's palate is thin/watery or thick/syrupy, or if it's under- or over-carbonated, that can affect how the taste is perceived and how the beer is rated overall. The negative interaction may help control for some of the positive correlation between taste and palate.

ABV starts to have a significant effect on the overall rating when increased by 5-10% which is a big shift. Alcohol content generally has a negative effect on overall rating after controlling for other factors. It interacts positively with taste and negatively with palate, but there is no scenario where the trend flips from negative to positive for overall rating vs. ABV. This is a bit surprising and warrants further investigation in the future. Generally, it seems people like higher alcohol beers as long as they're balanced and not too "hot" on the palate. Perhaps certain beer styles are masking this trend.

For any future modeling efforts, methods that account for multicollinearity should be used such as partial least squares or principal factor analysis. Ordinal logistic regression could also be used to model the probabilities of the discrete individual ratings.

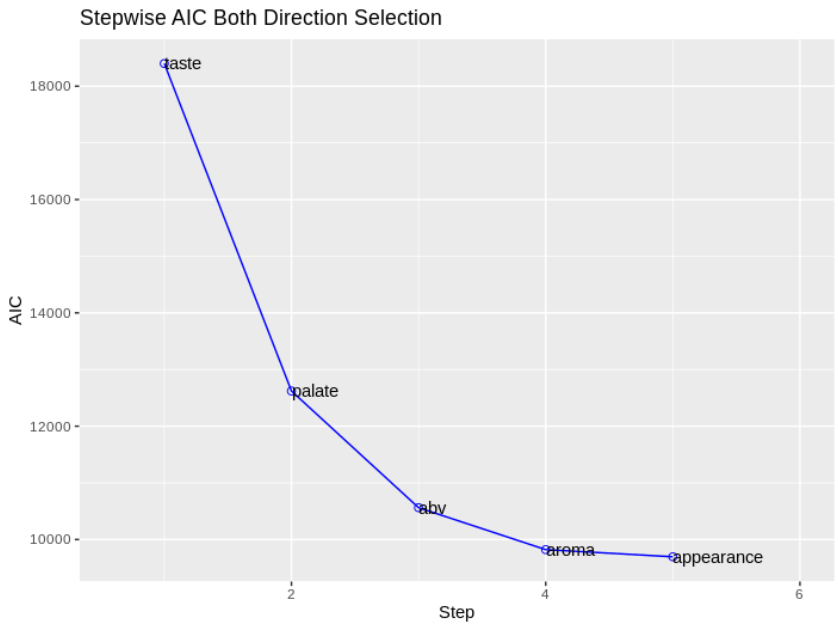

Figure IV.6 – Stepwise AIC plot of showing the important features

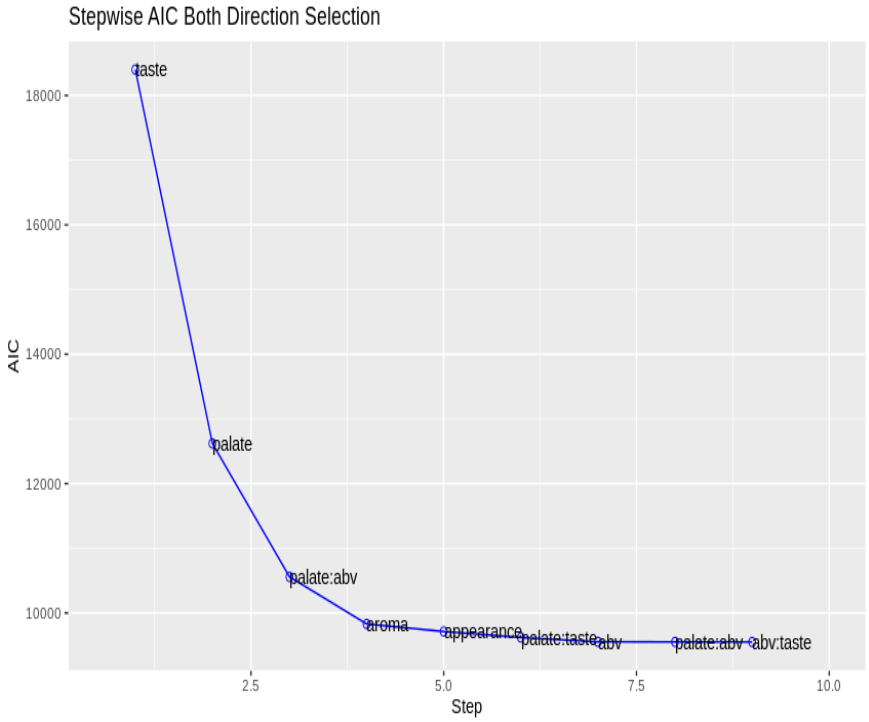

Figure IV.7 – Stepwise AIC plot of showing the important features including the interaction features

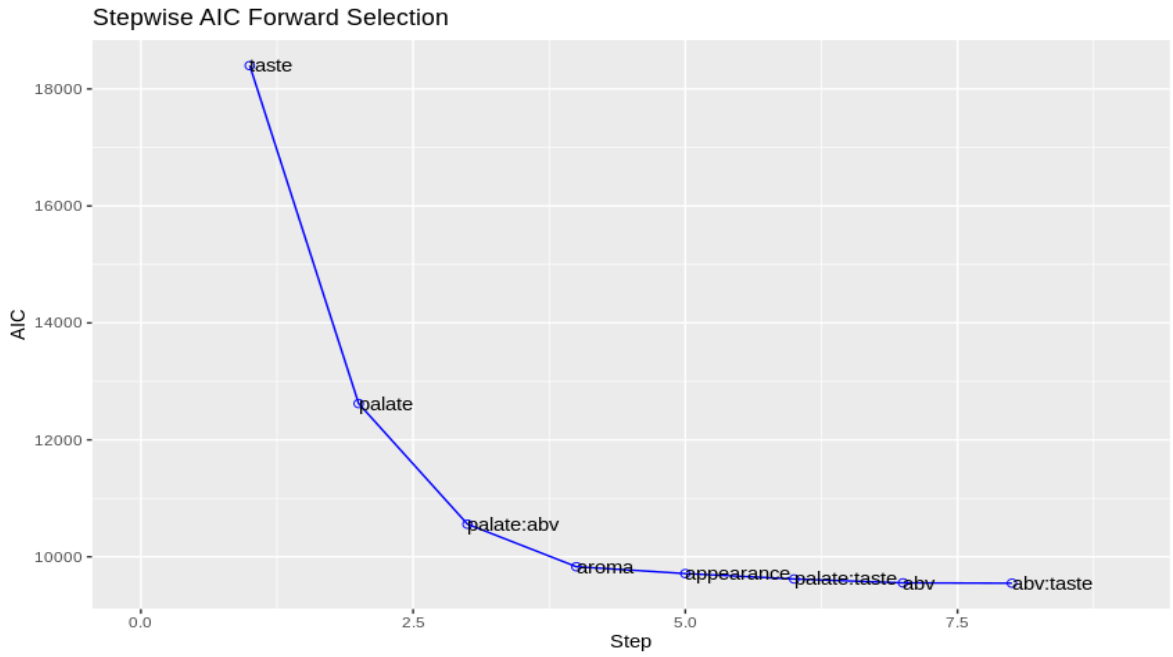

Figure IV.8 – Forward AIC plot showing important features including the interaction features

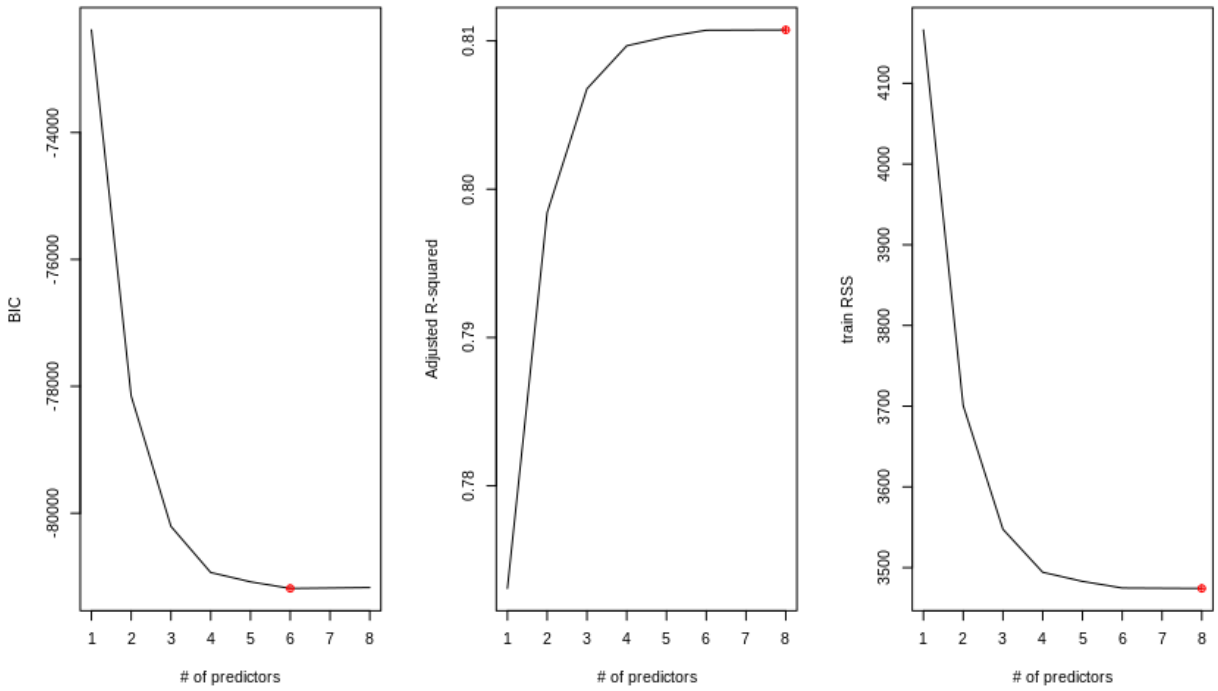

Figure IV.9 – Model selection for BIC, Adjusted R-Square, RSS

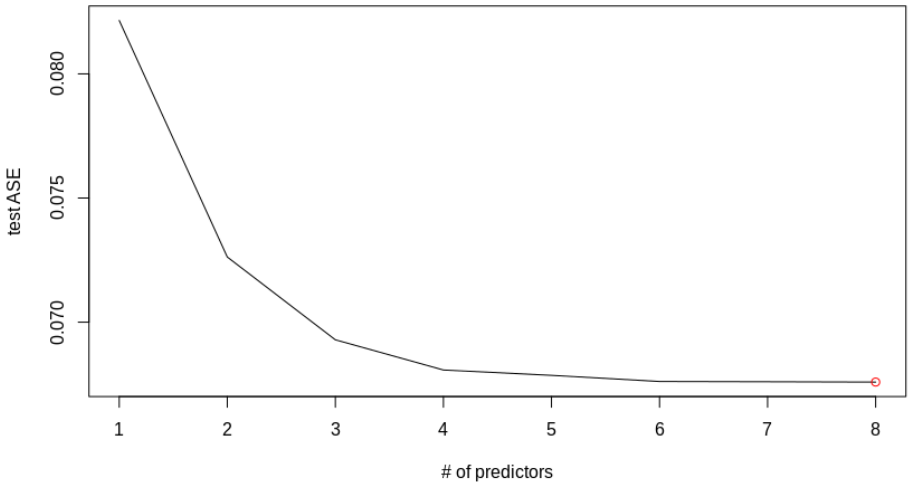

Figure IV.10 – Model selection for ASE

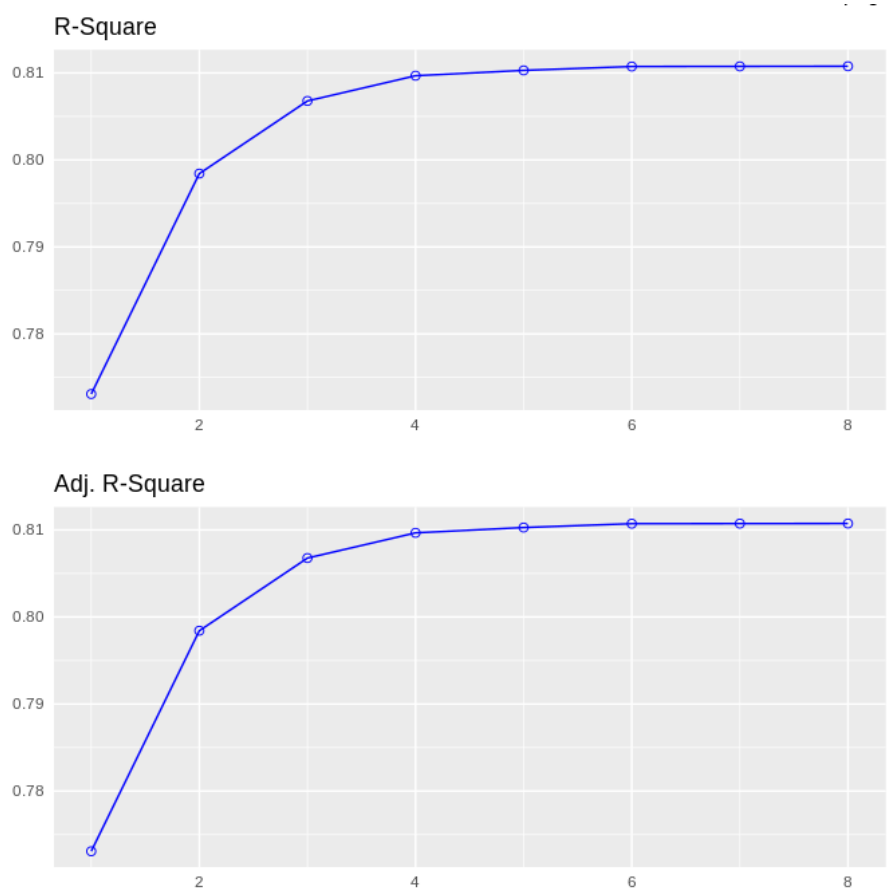

Figure IV.11 - R-Square and Adj R-Square plots

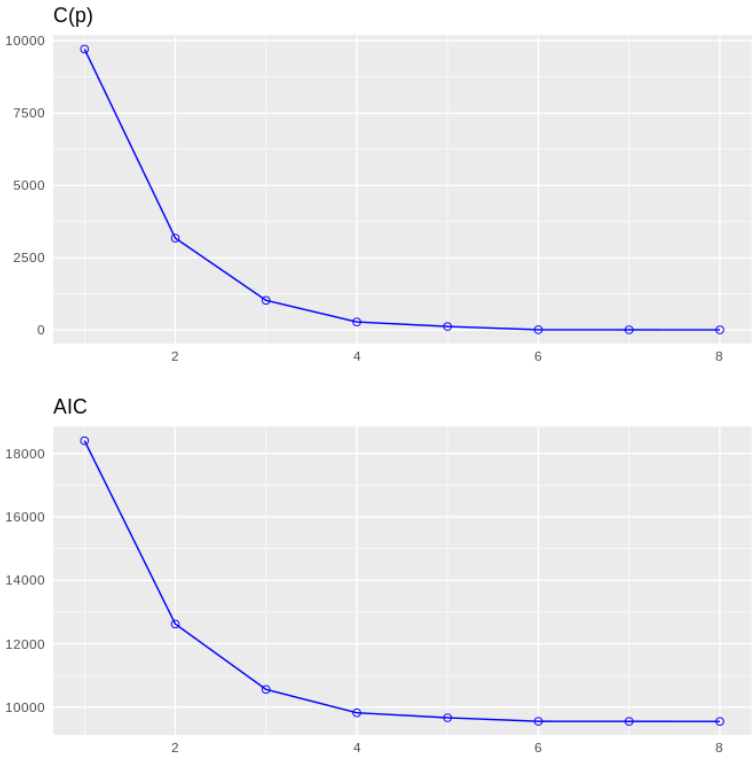

Figure IV.12 - C(p) and AIC plots

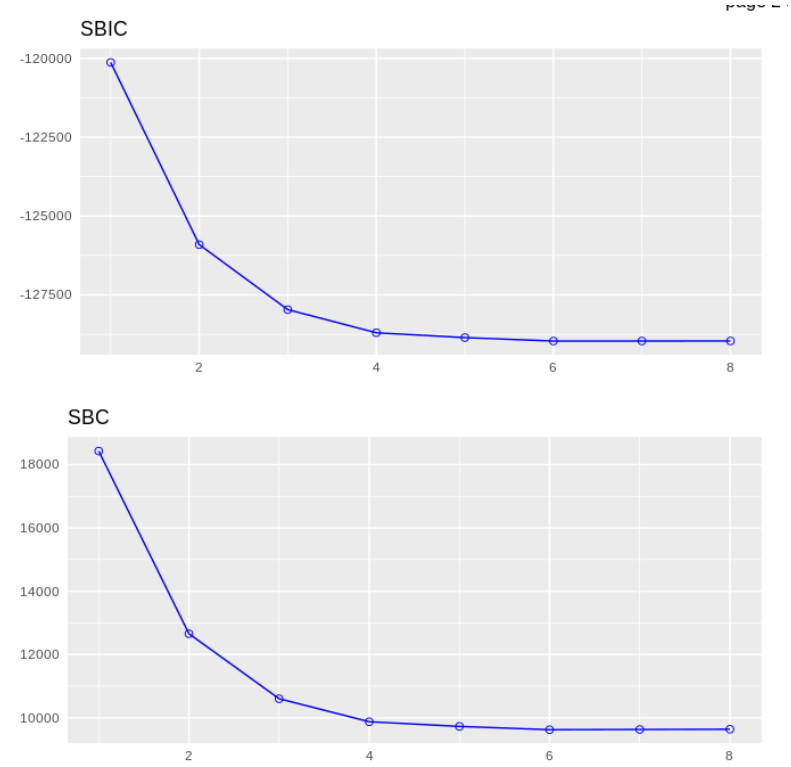

Figure IV.13 - SBIC and SBC plots

V. Objective 2

Goal Summary

Time Series Analysis (*cf.* [3,7,8,10]) is a statistical technique that deals with time series data, or trend analysis. We will perform a secondary analysis using Time Series and address if the assumption of independent errors is valid for the final regression model. The study is also to determine if there are any meaningful time-based trends in the beer review data.

Main Analysis Content

To reiterate from the Exploratory Analysis section, unequal data spacing over time and normality were addressed by averaging the beer reviews by Month. Average monthly review data for Dogfish Head brewery's "90 Minute IPA" look fairly stationary (Figure III.6). However, the first seven months are highly variable compared to the rest of the data. This is likely a violation of the constant variance assumption.

After some investigation, these early months have low sample sizes (n<7) from when the beer was first released with very limited distribution. These first points were filtered out because of the insufficient sample size and high variance, so any further analysis will only be applicable for August 2002 onwards. The filtered data vs. time are shown in Figure V.1 below.

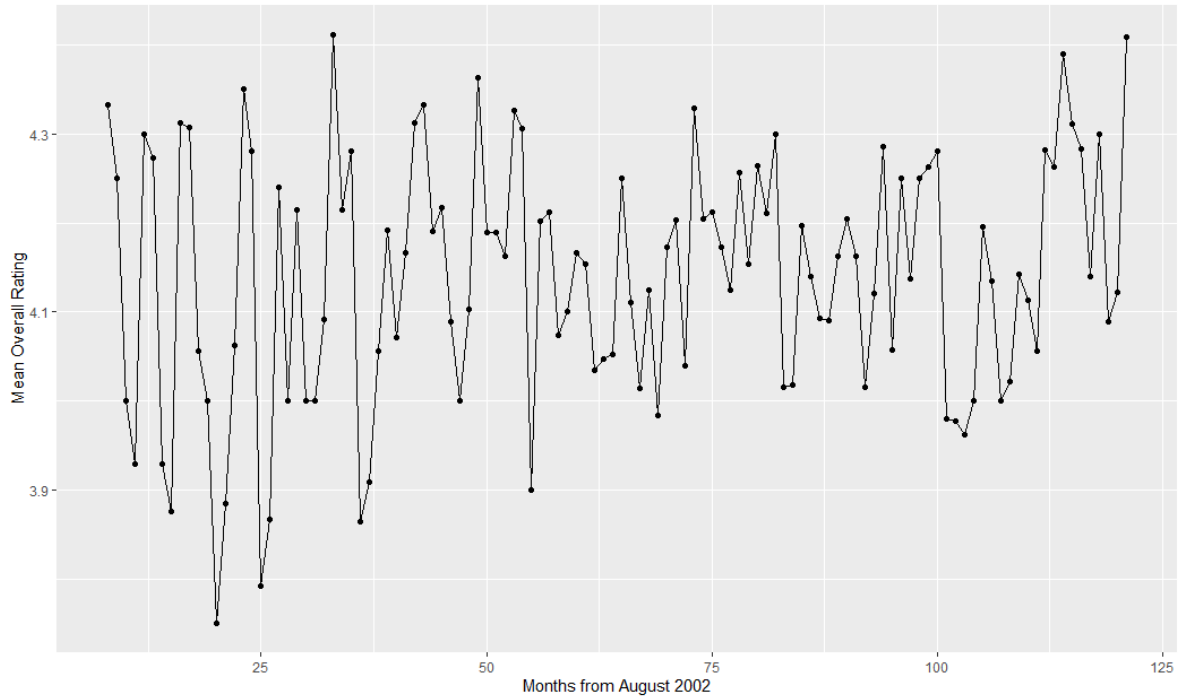

Figure V.1 - Average Overall Reviews for beer "90 Minute IPA" by Month from August 2002. Data appear to be stationary with approximately constant variance.

The ACF and PACF plots below for the filtered data indicate a possible ARMA model with both the ACF and PACF plots slowly decaying over time. There may be a significant AR(1) model per the ACF plot and/or an MA(2) model per the PACF plot. There may also be some higher order behavior at lag 13.

Figure V.2 - ACF and PACF plots for all average monthly reviews of Dogfish Head's 90 Minute IPA from August 2002 to late 2011.

The auto.arima function in R fit a MA(1) model that captures the frequency of the data with a slight forward offset, but the magnitude of the monthly changes is dampened (Figure V.3).

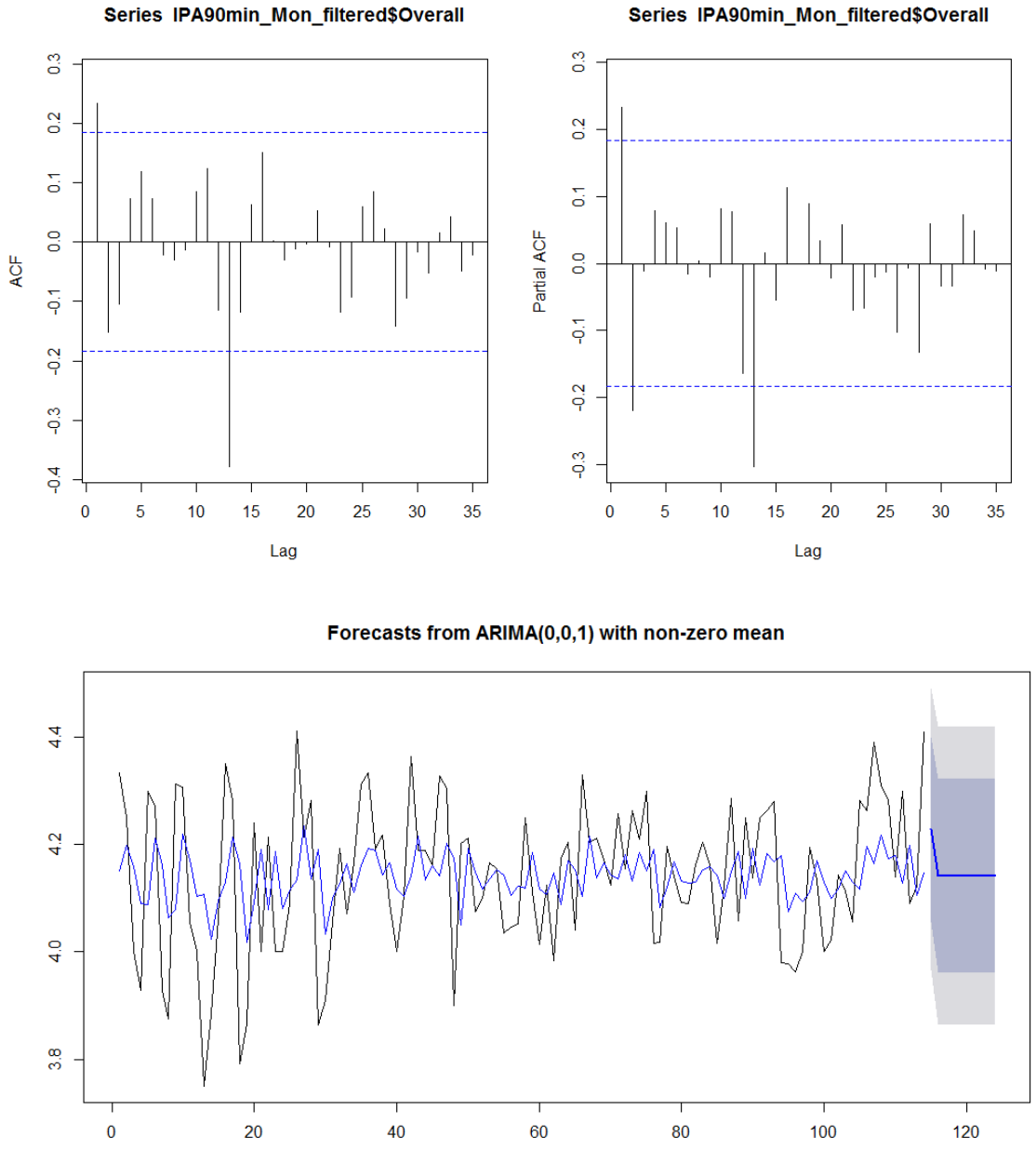

Figure V.3 – MA(1) time series fit and forecast for 90 Minute IPA Overall Rating by Month from August 2002 to late 2011.

Also, there are significant negative residuals at lag 13 for both ACF and PACF plots (Figure V.4) which corresponds to a 1 year plus 1 month lag. This is an indication that the errors are not independent for this model. Some hypotheses for this significant residual lag are:

- Users can't see previous reviews that are more than a year old on the forum which shifts the reviews
- Dogfish Head adjusts their recipe each year due to natural variation in the grain and hop harvest
- It may take 13 months on average for old product to cycle out of the market. IPAs are generally best when fresh.

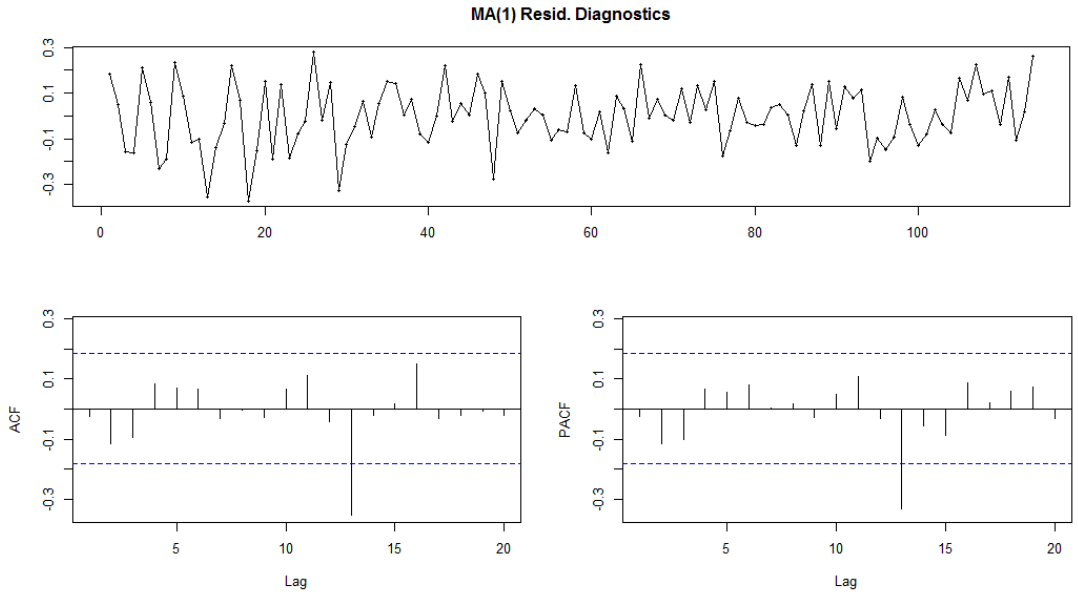

Figure V.4 – MA(1) time series residuals for "90 Minute IPA" Overall Rating by Month from August 2002 to late 2011. Errors are not independent due to significant lag 13 residuals.

Several attempts were made to improve the time series model to meet the independent errors assumption. This was done by adjusting the auto.arima function to generate a higher order model and by manually tuning the model. These models are summarized in Table V.1.

Table V.1 – Time Series Model Iterations for monthly reviews of "90 Minute IPA".

Model	**Selection Criteria**	**Independent Errors?**	**AIC**	**RMSE**
MA(1)	Auto, AIC, Stepwise	No, Lag 13	-130.4	0.133
AR(2)	Auto, AIC, Non-Stepwise	No, Lag 13	-131.2	0.131
AR(13)	Manual	Yes	-131.9	0.117
MA(13)	Manual	Yes	-134.6	0.113
AR(2,13)	**Manual**	**Yes**	**-148.8**	**0.119**
MA(1,13)	Manual	Yes	-150.5	0.118
AR(1)MA(2,13)	Manual	Not invertible	-151.6	0.109

The final model selected was the manually tuned AR(2,13) model that accounts for the lag 13 residual while balancing bias vs. variance. The coefficients between lags 3 and 13 were manually set to zero to avoid over-fitting (reduce variance). For example, the coefficients of the AR(13) model for lags 3-12 are likely not significant based on their value relative to their standard error (s.e.):

Table V.2 – AR(13) Model Coefficients and standard errors with likely insignificant terms greyed out.

	ar1	ar2	ar3	ar4	ar5	ar6	ar7	ar8
	0.2322	-0.1202	-0.0036	0.0398	0.0322	0.0656	0.0138	0.0361
s.e.	0.0874	0.0903	0.0900	0.0915	0.0915	0.0927	0.0917	0.0940

	ar9	ar10	ar11	ar12	ar13	intercept
	-0.0262	0.0351	0.0774	-0.0992	-0.4011	4.1401
s.e.	0.0941	0.0952	0.0954	0.0950	0.0952	0.0103

In contrast to that, the coefficients for the final model are:

Table V.3 –AR(1)MA(2,13) Model Coefficients and standard errors, likely all significant.

	ar1	ar2	ar3	ar4	ar5	ar6	ar7	ar8	ar9	ar10	ar11	ar12	ar13	intercept
	0.2331	-0.1387	0	0	0	0	0	0	0	0	0	0	-0.4392	4.1391
s.e.	0.0848	0.0851	0	0	0	0	0	0	0	0	0	0	0.0913	0.0086

Based on the residual plot for the final model (Figure V.5), the assumption of independent errors has been met.

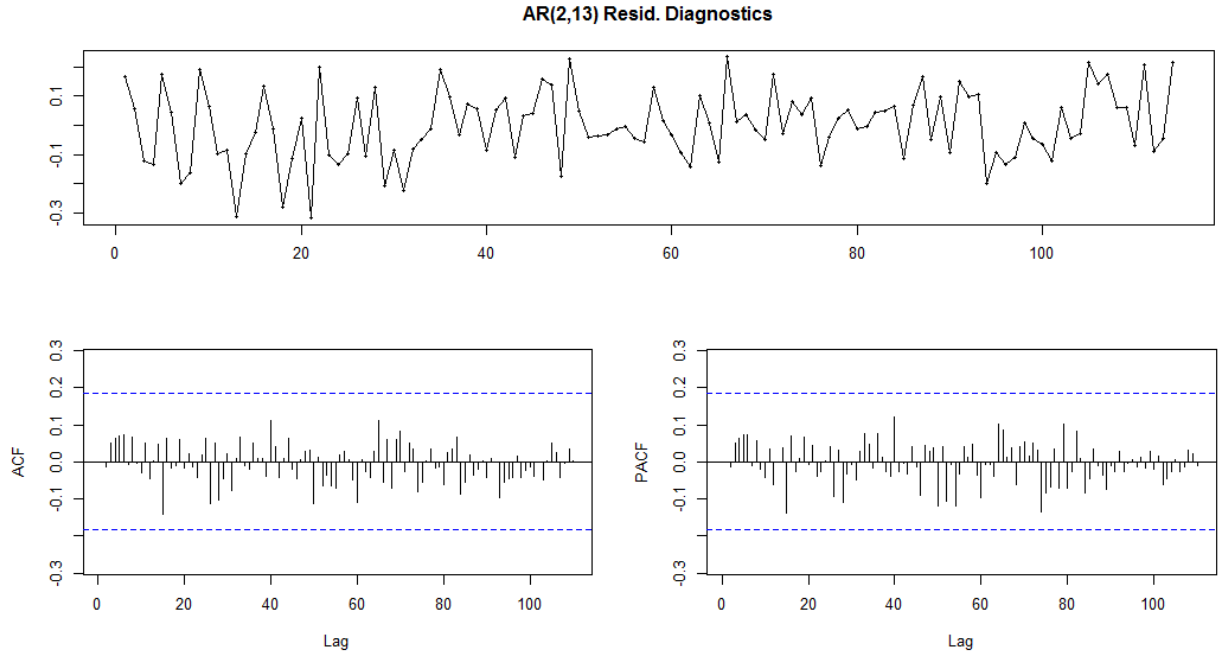

Figure V.5 – AR(2,13) time series residuals for "90 Minute IPA" Overall Rating by Month from August 2002 to late 2011. Errors are independent.

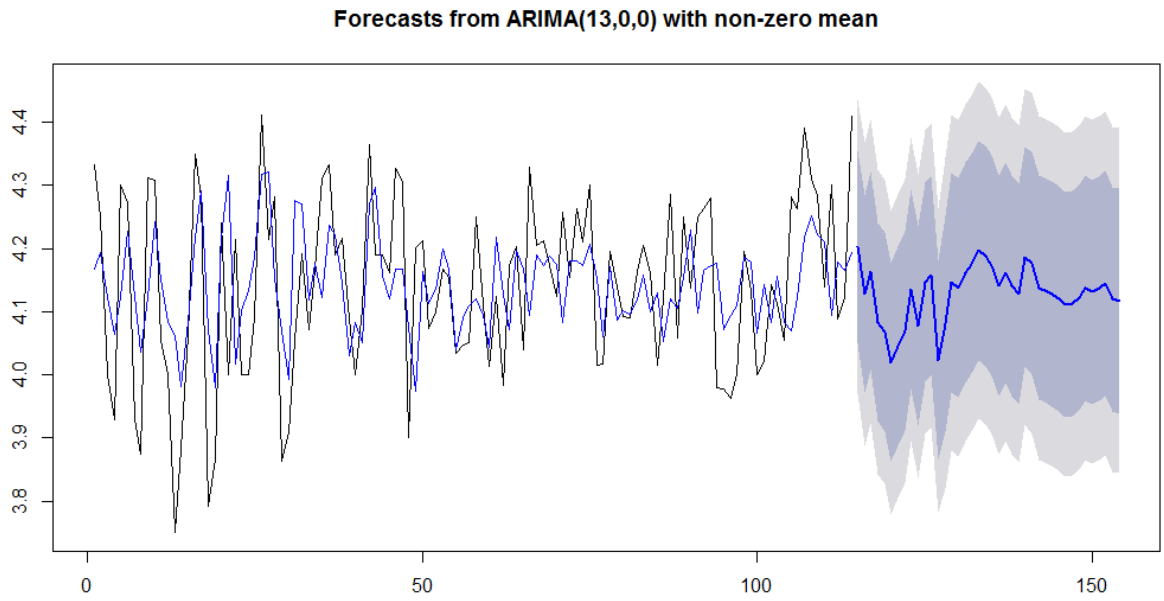

Figure V.6 – Final AR(2,13) time series fit and forecast for 90 Minute IPA Overall Rating by Month from August 2002 to late 2011.

Conclusion/Discussion

In conclusion, a time series model was developed for the average monthly beer reviews of DogFish Head brewery's "90 Minute IPA" from August 2002 to late 2011. Monthly average reviews depended on the previous 2 months as well as the 13th previous month, and once these trends were modeled the errors were independent over time. This 13th month previous trend is an interesting finding that warrants further investigation into whether it is related to the review system, the beer supply chain, or something else.

REFERENCES

[1] *Beer Advocate dataset* https://www.beeradvocate.com

[2] S. C. Albright, W. L. Winston, *Business Analytics - Data Analysis and Decision Making*, 7th Edition, Cengage, 2019.

[3] D. R. Anderson et al, *Statistics for Business & Economics*, 14th Edition, Cengage, 2020.

[4] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*, 2th Edition, Springer, 2017.

[5] G. James, D. Witten, T. Hastie, R. Tibshirani*, An Introduction to Statistical Learning with Application in R*, Springer, 2017.

[6] B. W. Lindgren, *Statistical Theory*, 3th Edition, MacMillan Publishing, 1976.

[7] D. Montgomery, E. A. Peck, G. G. Vining, *Introduction to Linear Regression Analysis*, 5th Edition, John Wiley & Sons, 2012.

[8] R. Peck, T. Short, C. Olsen, *Introduction to Statistics & Data Analysis*, 6th Edition, Cengage, 2020.

[9] F. L. Ramsey, D. W. Schafer, *The Statistical Sleuth - A Course in Methods of Data Analysis*, 3th Edition, Cengage, 2018.

[10] R. H. Shumway, D. S. Stoffer, *Time Series Analysis and Its Applications with R Examples*, 4th Edition, Springer, 2017.

ANDREW LEPPLA – Southern Methodist University – *Email:* aleppla@smu.edu

HUY HOANG NGUYEN - Southern Methodist University - *Email address:* hoangnguyen@smu.edu

IKENNA NWAOGU - Southern Methodist University - *Email address:* inwaogu@smu.edu

APPENDIX

Objective 1 - SAS code:

Import and Inspect Beer Reviews Data

```
proc import datafile= '/folders/myshortcuts/SASUniversityEdition/DS6372/files/proj1.csv'
        dbms=dlm out=beer replace;
    delimeter=',';
    getnames=yes;
run;
```

Scatterplot Matrix of Beer Variables

```
proc sgscatter data=beer;
 title "Scatterplot Matrix of Beer Variables";
 matrix all aroma appearance palate abv taste count;
run;
```

Add new column

```
data beer;
set beer;
logAll=log(all);
run;
```

Build regression models

```
proc reg data = beer;
model all =  aroma appearance palate abv taste count;
run;
proc reg data = beer;
model logAll = abv taste;
run;
proc reg data = beer;
model All = abv taste;
run;
proc reg data = beer;
model all =  aroma appearance palate abv taste; run;
```

Objective 1 - R Code:

Used libraries

```
library(dplyr)

library(plotly)

library(leaps)

library(MASS)

library(caret)

library(olsrr)

library(car)
```

Import and Inspect Beer Reviews Data

```
setwd("~/datascience/DS6372/ProjectDetails")

reviews<-read.csv("beer_reviews.csv")

names(reviews)
```

Aggregate mean reviews

```
regData <- reviews %>% filter(!is.na(beer_abv)) %>%
group_by(brewery_name,beer_name) %>% summarise(all =
mean(review_overall),aroma = mean(review_aroma), appearance =
mean(review_appearance), palate = mean(review_palate),abv = mean(beer_abv),
taste = mean(review_taste),count = n())

names(regData)

pairs(regData[3:9]) # this is to compare all the points including count(no of
reviews per beer)

hist(regData$count)
```

Build models

```
# Set seed for reproducibility

set.seed(124)
```

```r
model7 = lm(all ~ aroma + appearance + palate + abv + taste, data = regData)

summary(model7)

k <- ols_step_both_aic(model7)

plot(k)

full.model <- lm(all ~ aroma + appearance + palate + abv + taste  + abv*taste
+ abv*palate + palate* taste,data=regData)


## All possible regression model and the values they provide.

all <- ols_step_all_possible(full.model)

#View(all)

plot(all) #This displays all the attributes of the all data into a plot
showing which ones are the better values.

# Best subset of regression model

subset <- ols_step_best_subset(full.model)

subset   # This shows all the best model based on the values of a given
parameter

plot(subset)  # This displays all the parameters on a plot

# AIC forward

forw <- ols_step_forward_aic(full.model)

plot(forw)

#AIC backward

bac <- ols_step_backward_aic(full.model)

plot(bac)

#AIC both

k <- ols_step_both_aic(full.model)

plot(k)

# Stepwise regression model for AIC

step.model <- stepAIC(full.model, direction = "both",
```

```
                      trace = FALSE)

summary(step.model)    # The stars in the parenthesis means that it is
involved in the model.


# forward regression model for AIC

step.model <- stepAIC(full.model, direction = "forward",

                      trace = FALSE)

summary(step.model)


# Backward regression model For AIC

step.model <- stepAIC(full.model, direction = "backward",

                      trace = FALSE)

summary(step.model)


model1 = lm(all ~ aroma + appearance + palate + abv + taste + abv*palate +
abv*taste + taste*palate, data = regData)

confint(model1)


# Model selection using regsusets. We can use below to support the subset of
the earlier model

# selection using stepwise

models <- regsubsets(all ~ aroma + appearance + palate + abv + taste  +
abv*taste + abv*palate + palate* taste,data=regData, nvmax = 8,

                     method = "seqrep")

summary(models)


# Model Selection using forward

models <- regsubsets(all ~ aroma + appearance + palate + abv + taste  +
abv*taste + abv*palate + palate* taste,data=regData, nvmax = 8,
```

```
                    method = "forward")
summary(models)


# Model Selection using Backward
models <- regsubsets(all ~ aroma + appearance + palate + abv + taste  +
abv*taste + abv*palate + palate* taste,data=regData, nvmax = 8,
                    method = "backward")
summary(models)
# Model selection using K-fold cross validation and train() of the caret
package
# Set up repeated k-fold cross-validation
train.control <- trainControl(method = "cv", number = 10)
# Train the model using backward
step.model <- train(all ~ aroma + appearance + palate + abv + taste  +
abv*taste + abv*palate + palate* taste,data=regData,
                    method = "leapBackward",
                    tuneGrid = data.frame(nvmax = 1:8),
                    trControl = train.control
)
step.model$results
step.model$bestTune
summary(step.model$finalModel)


# Train the model using forward
step.model <- train(all ~ aroma + appearance + palate + abv + taste  +
abv*taste + abv*palate + palate* taste,data=regData,
                    method = "leapForward",
                    tuneGrid = data.frame(nvmax = 1:8),
```

```
                    trControl = train.control

)

step.model$results


step.model$bestTune


summary(step.model$finalModel)

# Train the model using stepwise

step.model <- train(all ~ aroma + appearance + palate + abv + taste  +
abv*taste + abv*palate + palate* taste,data=regData,

                    method = "leapSeq",

                    tuneGrid = data.frame(nvmax = 1:8),

                    trControl = train.control

)

step.model$results

step.model$bestTune

summary(step.model$finalModel)

# As you can see each and every model selection gave us different models look
at those and let me know what you think. I think either one of 5,7,8 is good.
```

Objective 2 - R Code:

Import and Inspect Beer Reviews Data

```
beer_reviews = read.csv('~/Documents/R/Stats2/beer_reviews.csv')
#head(beer_reviews)
#str(beer_reviews)
```

Recode review_time variable for Time Series

```
library(dplyr)

library(tidyverse)

library(ggplot2)

#Investigate review_time for recoding
max_time = max(beer_reviews$review_time) #November 2011?
min_time = min(beer_reviews$review_time)
diff = (max_time- min_time)/60/60/24/365.25 #Over 10 years?
diff #Actually over 15 years

## [1] 15.38816

#Recode review_time with the common origin time Jan 1, 1970
beer_reviews$review_date_time =
as.POSIXct.numeric(beer_reviews$review_time,origin='1970-01-01
00:00:00',tz='EST')
head(beer_reviews$review_date_time)

## [1] "2009-02-16 15:57:03 EST" "2009-03-01 08:44:57 EST"
## [3] "2009-03-01 09:10:04 EST" "2009-02-15 14:12:25 EST"
## [5] "2010-12-30 13:53:26 EST" "2012-01-02 12:17:39 EST"

max(beer_reviews$review_date_time) #Jan 11, 2012

## [1] "2012-01-11 07:35:48 EST"

min(beer_reviews$review_date_time) #Aug 21, 1996

## [1] "1996-08-21 19:00:01 EST"

##############################################################################
#Explore Overall reviews vs. time for 90 Minute IPA (most-rated beer)
beer_reviews %>% filter(review_date_time>'2011-01-01 00:00:00' &
beer_name=='90 Minute IPA') %>%
ggplot(aes(x=review_date_time,y=review_overall,color=beer_name)) +
geom_line()

#Data is not equally spaced over time (as expected)
##############################################################################

#Average by month for equally spaced data over time
```

```
beer_reviews$time_vectors =
as.POSIXlt.numeric(beer_reviews$review_time,origin='1970-01-01
00:00:00',tz='EST') #Splits the time "YYYY-MM-DD HH:MM:SS" into vectors
"YYYY","MM","DD",etc.

beer_reviews$Year = as.numeric(beer_reviews$time_vectors$year) + 1900
#Calling $year from class POSIXlt resets the origin to year 1900 (see
?POSIXlt)
beer_reviews$Mon = as.numeric(beer_reviews$time_vectors$mon) + 1 # $mon from
POSIXlt is from 0-11, add 1 so it's from 1-12
```

Monthly Time Series - 90 Minute IPA

```
library(tseries)

library(forecast)

#Need to recode the time variable back to POSIXct to pipe it through
group_by() and summarize()
beer_reviews$time_vectors =
as.POSIXct.numeric(beer_reviews$review_time,origin='1970-01-01
00:00:00',tz='EST')

#Filter for 90 Minute IPA Average Monthly Reviews
tseries_90minIPA_Mon = beer_reviews %>% filter(beer_name=='90 Minute IPA')
%>% group_by(Year, Mon) %>% summarize(Overall=mean(review_overall),Count=n())
#class=tibble
tseries_90min_Mon_df=data.frame(tseries_90minIPA_Mon) #convert tibble to data
frame

#Plot the data before modeling
##Data has nonconstant variance for months 1-7 with low n
tseries_90min_Mon_df$Index=1:nrow(tseries_90min_Mon_df)
tseries_90min_Mon_df %>% ggplot(aes(x=Index,y=Overall)) + geom_line() +
geom_point() + ylab("Mean Overall Rating") + xlab("Months from Dec. 2001")

#ACF and PACF Plots
par(mfrow=c(1,2))
Acf(tseries_90min_Mon_df$Overall,lag.max=30) #Up to lag 19 looks significant
Pacf(tseries_90min_Mon_df$Overall,lag.max=30) #Up to lag 19 looks significant

par(mfrow=c(1,1))
```

Monthly Time Series - Filter out the first 7 months for constant variance

```
#Plot that filters out the first 7 months with
#nonconstant variance and insufficient sample sizes
tseries_90min_Mon_df %>% filter(Index>7) %>% ggplot(aes(x=Index,y=Overall)) +
```

```
geom_line() + geom_point() + ylab("Mean Overall Rating") + xlab("Months from
August 2002")

IPA90min_Mon_filtered = tseries_90min_Mon_df %>% filter(Index>7)
min(IPA90min_Mon_filtered$Year) #August 2002

## [1] 2002

par(mfrow=c(1,2))
Acf(IPA90min_Mon_filtered$Overall,lag.max=35) #Has a longer lag = 13
Pacf(IPA90min_Mon_filtered$Overall,lag.max=35)#Has a longer lag = 13

par(mfrow=c(1,1))


#Auto ARIMA

ARIMA.1<-auto.arima(IPA90min_Mon_filtered$Overall)
summary(ARIMA.1) #MA(1), missing lag 13, AIC = -130.2

plot(forecast(ARIMA.1,h=10))
points(1:length(IPA90min_Mon_filtered$Overall),fitted(ARIMA.1),type="l",col="
blue")

tsdisplay(residuals(ARIMA.1),lag.max=20,main="MA(1) Resid. Diagnostics")

tsdisplay(residuals(ARIMA.1),lag.max=115,main="MA(1) Resid. Diagnostics")

ARIMA.2<-auto.arima(IPA90min_Mon_filtered$Overall,stepwise=F)
summary(ARIMA.2) #AR(2), missing lag 13, AIC = -131.2

plot(forecast(ARIMA.2,h=10))
points(1:length(IPA90min_Mon_filtered$Overall),fitted(ARIMA.2),type="l",col="
blue")

tsdisplay(residuals(ARIMA.2),lag.max=20,main="AR(2) Resid. Diagnostics")

#Manual ARIMA

MA_13 = arima(IPA90min_Mon_filtered$Overall, order=c(0,0,13))
summary(MA_13) #AIC = -134.6

plot(forecast(MA_13,h=20))
points(1:length(IPA90min_Mon_filtered$Overall),fitted(MA_13),type="l",col="bl
ue")

tsdisplay(residuals(MA_13),lag.max=110,main="MA(13) Resid. Diagnostics")

AR_13 = arima(IPA90min_Mon_filtered$Overall, order=c(13,0,0))
summary(AR_13) #AIC = -131.9
```

```r
plot(forecast(AR_13,h=70))
points(1:length(IPA90min_Mon_filtered$Overall),fitted(AR_13),type="l",col="blue")

tsdisplay(residuals(AR_13),lag.max=110,main="AR(13) Resid. Diagnostics")



#Set insignificant coefficients to 0 with fixed and transform.pars=F (see ?arima)

MA_1_13 = arima(IPA90min_Mon_filtered$Overall, order=c(0,0,13), fixed=c(NA,0,0,0,0,0,0,0,0,0,0,0,NA,NA),transform.pars=F)
summary(MA_1_13) #AIC = -150.5

plot(forecast(MA_1_13,h=20))
points(1:length(IPA90min_Mon_filtered$Overall),fitted(MA_1_13),type="l",col="blue")

tsdisplay(residuals(MA_1_13),lag.max=110,main="MA(1,13) Resid. Diagnostics")

AR_2_13 = arima(IPA90min_Mon_filtered$Overall, order=c(13,0,0),fixed=c(NA,NA,0,0,0,0,0,0,0,0,0,0,0,NA,NA),transform.pars=F)
summary(AR_2_13)

plot(forecast(AR_2_13,h=40))
points(1:length(IPA90min_Mon_filtered$Overall),fitted(AR_2_13),type="l",col="blue")

tsdisplay(residuals(AR_2_13),lag.max=110,main="AR(2,13) Resid. Diagnostics")

#Model with best AIC and RMSE - NOT INVERTIBLE!!!
AR_1_MA_2_13 = arima(IPA90min_Mon_filtered$Overall, order=c(13,0,13),fixed=c(NA,0,0,0,0,0,0,0,0,0,0,0,0,0,NA,NA,0,0,0,0,0,0,0,0,0,0,0,NA,NA),transform.pars=F)
summary(AR_1_MA_2_13)

plot(forecast(AR_1_MA_2_13,h=30)) #NOT INVERTIBLE

## Warning in predict.Arima(object, n.ahead = h): MA part of model is not
## invertible

points(1:length(IPA90min_Mon_filtered$Overall),fitted(AR_1_MA_2_13),type="l",col="blue")

tsdisplay(residuals(AR_1_MA_2_13),lag.max=110,main="AR(1)MA(2,13) Resid. Diagnostics")
```

##Differencing

These models don't capture the significant lag 13 residuals and may be overfitting the data.

```
#Differencing degree of 1 (d=1)
#Stationary model per the forecast plot
ARIMA.d1<-auto.arima(IPA90min_Mon_filtered$Overall,d=1)
summary(ARIMA.d1)

plot(forecast(ARIMA.d1,h=15))
points(1:length(IPA90min_Mon_filtered$Overall),fitted(ARIMA.d1),type="l",col=
"blue")

tsdisplay(residuals(ARIMA.d1),lag.max=20,main="MA(1) Resid. Diagnostics")

#Differencing degree of 2 (d=2)
#Nonstationary model per the forecast plot
ARIMA.d2<-auto.arima(IPA90min_Mon_filtered$Overall,d=2,stepwise=F)
summary(ARIMA.d2)

plot(forecast(ARIMA.d2,h=15))
points(1:length(IPA90min_Mon_filtered$Overall),fitted(ARIMA.d2),type="l",col=
"blue")

tsdisplay(residuals(ARIMA.d1),lag.max=20,main="MA(1) Resid. Diagnostics")

#Manual Differencing
##This appears to be doing 13 degrees of differencing rather than
differencing for lag 13

diff.data<-arima(IPA90min_Mon_filtered$Overall,order=c(0,13,0))
summary(diff.data)

tsdisplay(residuals(diff.data),lag.max=30,main="Resid. Diagnostics 1st Order
Difference")

Diff1 = auto.arima(diff.data$residuals)
summary(Diff1) #AR(3)

plot(forecast(Diff1,h=10))
points(1:length(diff.data$residuals),fitted(Diff1),type="l",col="blue")

tsdisplay(residuals(Diff1),lag.max=105,main="AR(3) Resid. Diagnostics")

Diff2 = auto.arima(diff.data$residuals,stepwise=F)
summary(Diff2) #AR(1)MA(4)

plot(forecast(Diff2,h=10))
points(1:length(diff.data$residuals),fitted(Diff2),type="l",col="blue")

tsdisplay(residuals(Diff2),lag.max=105,main="AR(1)MA(4) Resid. Diagnostics")
```