Case Study: Modeling Liquid Mechanics

Conner Byrd, Eric Han, Ki Hyun, Sara Shao, Alex Shen, Mona Su, Dani Trejo, Steven Yuan

10/12/2021

Introduction

Our key research objectives include understanding and predicting how turbulence affects the dynamics of water droplets and ice crystals (how they collide and mix) in clouds. With our machine learning model, we are trying to infer the volume distribution of clusters within clouds. Our graphs have been included in the Appendix section of this report.

Methodology

Initial Approach

We decided at the beginning to treat `Fr` and `Re` as ordered factors. It seemed to make sense to treat Fr as a categorical variable because we are given only three unique values. In terms of prediction and inference, we believe our models can still be generalized to `Fr` and `Re` values similar to the ones we are working with.

```
first_lm_R1 <- lm(R_moment_1 ~ St + Re + Fr, data = train_data)
first_lm_R2 <- lm(R_moment_2 ~ St + Re + Fr, data = train_data)
first_lm_R3 <- lm(R_moment_3 ~ St + Re + Fr, data = train_data)
first_lm_R4 <- lm(R_moment_4 ~ St + Re + Fr, data = train_data)
summary(first_lm_R1)$r.squared
```

```
## [1] 0.9293093
```

```
summary(first_lm_R4)$r.squared
```

```
## [1] 0.4252642
```

We decided to initially fit the most basic linear model with all three predictors to see what it would look like. The model for the first moment had a fairly high R^2 value (0.929), but the model for the fourth moment had a much lower R^2 value (0.425). Additionally, in our diagnostic plots (see Appendix: Section 2), we saw a pattern in the residuals vs. fitted values plots where the models would consistently under predict in some areas and over predict in others, indicating non-linearity. In addition, looking at the Normal Q-Q plots, the normality assumption also seemed to be violated for higher moments. This is consistent with the fact that our histogram of `St` in our EDA was not normally distributed.

This information lead us to try using a GAM to model the relationship between the predictors and the 4 moments due to the increased flexbility GAMs provide. However, we knew that using GAMs made interpretability an issue, because interpreting a complex smooth function of a continuous predictor is very difficult

Variable transformation

As a result, we decided to use variable transformations and include interaction effects to make linear models with suitable model diagnostics for all 4 moments for the purpose of inference. To decide how to apply variable transformations, we observed the predictor and response variables, took a deeper look at the purpose of the study, and found three main problems.

1. Just converting `Fr`, and `Re` into categorical values may result in losing information as the exact numerical value could have high relationship with the moments. Furthermore, this model should be used later to predict behaviors of particles in environments with higher Re values.
2. The scale of `Fr`, `St`, and `Re` differs from each other. `Fr` ranges $[0, \infty)$, while `St` is $[0, 3]$. Furthermore, in the real world, `Re` could be in the scale of 10^7.
3. The response variables (moments) are heavily skewed. For example, for the 4th moment, some response variables are at the scale of 10^{-10}, while some are at the scale of 10^{10}

As a result, we considered transformations on the potential predictor variables. For Froude number (`Fr`), a monotonic transformation of $1 - e^{-Fr}$ was made to change the range from $[0, \infty)$ to $[0, 1]$. For Stokes number (`St`), as shown in the EDA (see Appendix: Section 1), log transformation was undertaken to address the skewness. For Reynolds number (`Re`), not only is the actual value in practice much larger than the given training data, but also within the training data itself, Reynolds number was much greater in scale. To address the difference in scale, log transformation was also made on Reynolds number.

```
train_data <- train %>%
    mutate(
        Re = log(Re),
        Fr = 1 - exp(-Fr),
        St = log(St)
    )
```

Transformation on the response variable also needed to take place before fitting the models. For the four moments, we have considered box-cox transformation to normalize the variables and adjust for the difference in scale. As the EDA shows (see Appendix: Section 1), the response variables have a skewed distribution. Thus, first, normalization was needed for the four moments. Moreover, the scale of the response variable becomes extremely larger as the order of the moments increase. (4th moment would have approximately to the power of 4 scale of the 1st moment) Hence, second, adjustments for scale were needed for the four moments. Box-cox transformation performs normalization and adjusts the scale of variables: apt for our response variable transformation.

However, the λ for box-cox transformation needed to be specified for each of the four moments. We have calculated and specified the four different lambdas as the maximum log-likelihood estimate as below.

```
lambda.1 = -0.0202
lambda.2 = -0.1010
lambda.3 = -0.0606
lambda.4 = -0.0606
```

Forward AIC for Linear Models

For linear model, we tried doing forward selection with AIC to see if we could select a simpler model in case were overfitting, but our resulting models were the same as our input models (see Appendix: Section 2). We would also compare our final linear models with 4 GAM models (one for each moment) using 10-fold CV in order to find the best models for prediction.

Cross Validation

For each moment, we considered various combinations of predictor variables and interactions of the transformed variables. In order to compare the models and choose the most optimal ones, we used 10-fold cross validation to estimate the test error. A major reason for choosing cross validation was that cross validation can provide a direct estimate for the test estimate, even when we do not have a clear picture of the noises of the variables. Furthermore, it is less likely to result in overfitting compared to LOOCV, and computationally more feasible.

To perform a valid 10-fold cross validation, we first shuffled the training dataset to ensure that similar data do not tend to be included in the same fold. Then, we divided the training dataset into 10 folds and used the same folds to perform cross validation and obtain estimated test error for each model. Below shows how the folds were created.

```
set.seed(3736)
shuffled_train <- train_data[sample(nrow(train_data)),]
folds <- cut(seq(1,nrow(train_data)),breaks=10,labels=FALSE)
```

At each fold, the validation set (8~9 data points for each fold) was reserved, and the other points were used to train a model. Then, the model was used to test on the validation set. RMSE was then calculated at each fold and the mean RSME was then calculated to compare the estimated test error with other model candidates. For each moment, the model with the lowest estimated test error was selected.

For each moment, the following linear models were considered to vary complexity and explore various possibilities:

1. GAM model without interaction
2. GAM models with one interaction term on each combination of predictor variables
3. GAM models with two interaction terms
4. GAM model with all possible interaction terms

Results

Final Linear Models

The final four linear models, one for each moment takes the form of the equations below.

- First Moment:

$$\frac{E(R)^{\lambda_1}-1}{\lambda_1} \sim (1-e^{-Fr})+log(St)+log(Re)+(1-e^{-Fr})\times log(St)+log(St)\times log(Re)+(1-e^{-Fr})\times log(Re)$$

- Second Moment:

$$\frac{E(R^2)^{\lambda_2}-1}{\lambda_2} \sim (1-e^{-Fr})+log(St)+log(Re)+(1-e^{-Fr})\times log(St)+log(St)\times log(Re)+(1-e^{-Fr})\times log(Re)$$

- Third Moment:

$$\frac{E(R^3)^{\lambda_3}-1}{\lambda_3} \sim (1-e^{-Fr})+log(St)+log(Re)+(1-e^{-Fr})\times log(St)+log(St)\times log(Re)+(1-e^{-Fr})\times log(Re)$$

- Fourth Moment:

$$\frac{E(R^4)^{\lambda_4}-1}{\lambda_4} \sim (1-e^{-Fr})+log(St)+log(Re)+(1-e^{-Fr})\times log(St)+log(St)\times log(Re)+(1-e^{-Fr})\times log(Re)$$

The model diagnostics plots of the four linear models shown in Appendix Figure 2.7-14. The independence condition for regression is thought to be met since there does not seem to be a reason to believe otherwise about the population. Moreover, from the diagnostics plots, the conditions for linearity and normality of the residuals seem to have met. There still may be some heteroscedasticity present in the residuals; however, compared to the initial linear models, the variable transformation seems to have made the residuals substantially more homoscedastic.

The full output of each final linear model is shown in Appendix Figure 2.6

The two predictors ($log(Re)$ and $1-e^{-Fr}$) and their interaction term seems to be statistically significant at $\alpha = 0.001$ level for all 4 final linear models. For the first moment, the interaction term between $log(St)$ and $1-e^{-Fr}$ also seems to be significant at $\alpha = 0.05$ level.

Final GAM Models As shown from the results of the 10-fold CV test error estimation in Appendix Figure 3.1, for each moment of R, the 6th model was selected as the final GAM model. The 6th model was configured with all three predictors and two interaction terms: one between $log(Re)$ and $1-e^{-Fr}$, and the other between $s(log(St))$ and $1-e^{-Fr}$. As indicated by the `s()`, smoothing function of GAM was applied to $log(St)$ to aid fitting the data better.

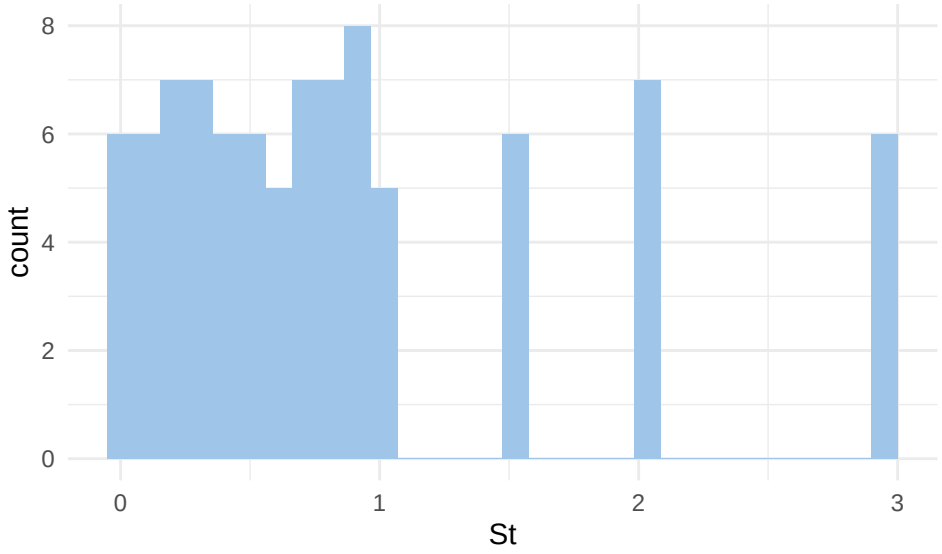
The model diagnostic plots of the four GAM models using function `gam.check()` showed that the conditions for the GAM model has met.

Though, due to the presence of a smoothing function, interpretation of the coefficients is challenging, GAM could still be an addition to the research if it brings more prediction power.

Test MSE Estimates: LM vs. GAM Comparing the estimated testing error between the 4 final LM models and the 4 final GAM models using 10-fold CV yielded that for the first moment the GAM model is preferred. For the second, third, and fourth moments, the linear models were preferred.

Appendix

Figure 1.1: Distribution of St



We will try using a log transform on the St variable since the distribution for the St variable is not normally distributed.

Figure 1.2: R Moments and Re Colored by Fr

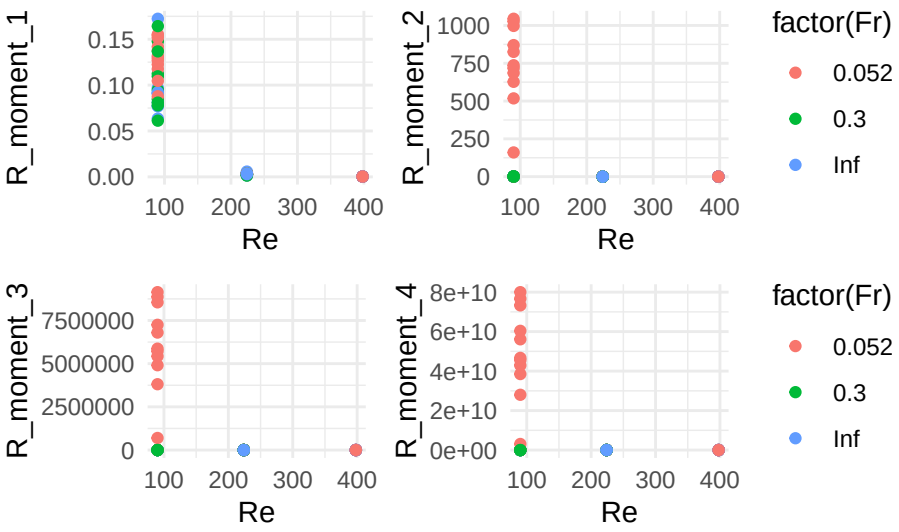
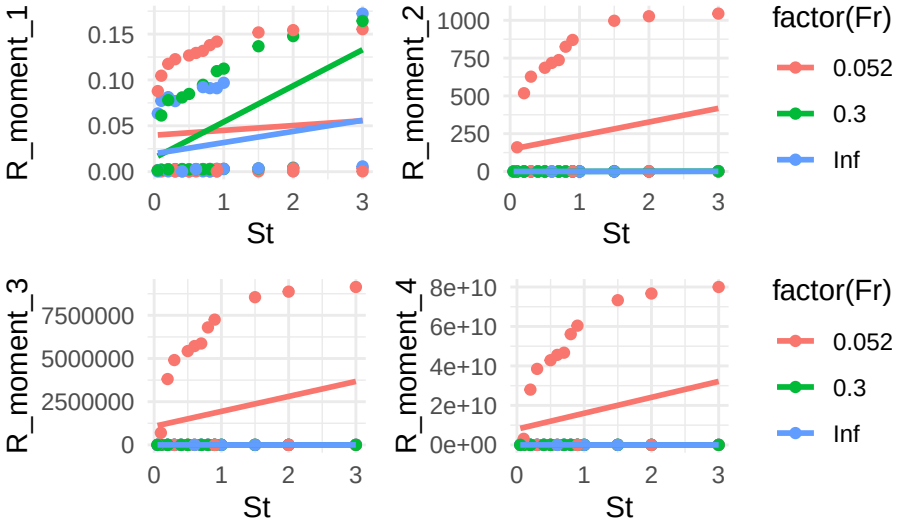
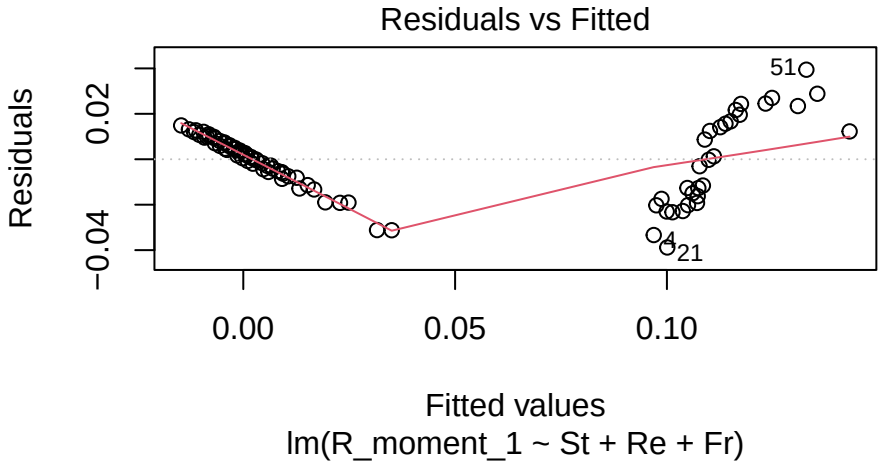


Figure 1.3: R Moments and St Colored by Fr



The graphs above show some evidence of interactions, so we will explore interaction terms in our model.

Figure 2.1-2: Diagnostic Plots For Inital Linear Model of R Moment 1



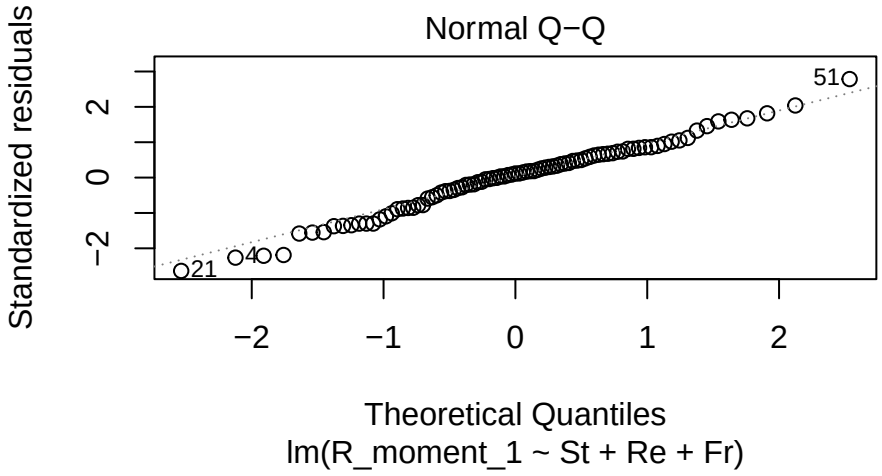
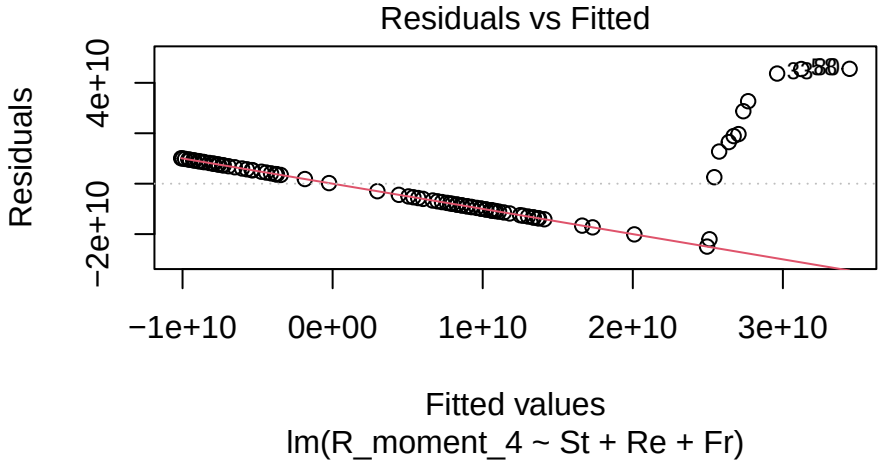
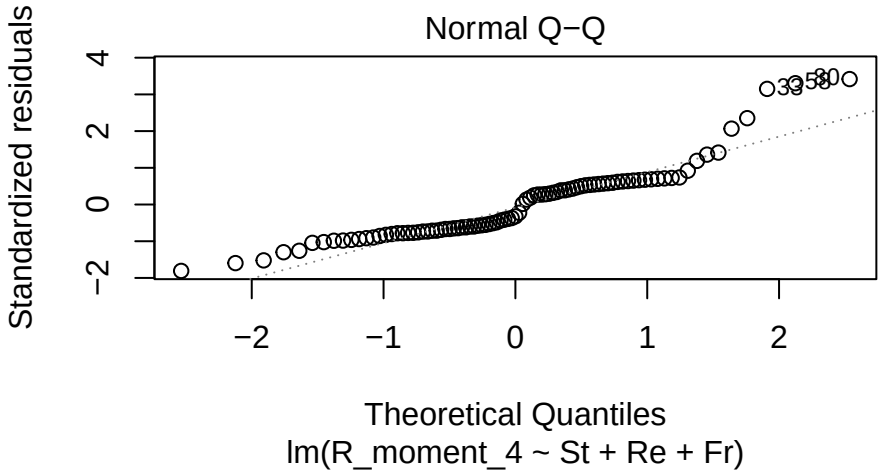


Figure 2.3-4: Diagnostic Plots For Inital Linear Model of R Moment 4





Because the linearity condition is not fulfilled in the above Residuals vs. Fitted plots, we will consider performing a log transformation on our response variables (R moments 1-4).

Figure 2.5: Forward Selection With AIC On Linear Models

```
lm1 <- lm((R_moment_1^lambda.1 - 1) /
            lambda.1 ~ St + Re + Fr + St*Fr + Fr*Re + St*Re, data = train_data)
step(lm1, direction = "forward")
```

```
## Start:  AIC=-362.41
## (R_moment_1^lambda.1 - 1)/lambda.1 ~ St + Re + Fr + St * Fr +
##     Fr * Re + St * Re

##
## Call:
## lm(formula = (R_moment_1^lambda.1 - 1)/lambda.1 ~ St + Re + Fr +
##     St * Fr + Fr * Re + St * Re, data = train_data)
##
## Coefficients:
## (Intercept)           St           Re           Fr        St:Fr        Re:Fr
##    17.90588      0.08917     -4.44003     -1.78892      0.06105      0.34834
##       St:Re
##     0.01950
```

```
lm2 <- lm((R_moment_2^lambda.2 - 1) /
            lambda.2 ~ St + Re + Fr + St*Fr + Fr*Re + St*Re, data = train_data)
#step(lm2, direction = "forward")

lm3 <- lm((R_moment_3^lambda.2 - 1) /
            lambda.3  ~ St + Re + Fr + St*Fr + Fr*Re + St*Re, data = train_data)
#step(lm3, direction = "forward")
```

```
lm4 <- lm((R_moment_4^lambda.4 - 1) /
            lambda.4 ~ St + Re + Fr + St*Fr + Fr*Re + St*Re, data = train_data)
step(lm4, direction = "forward")
```

```
## Start:  AIC=201.27
## (R_moment_4^lambda.4 - 1)/lambda.4 ~ St + Re + Fr + St * Fr +
##     Fr * Re + St * Re

##
## Call:
## lm(formula = (R_moment_4^lambda.4 - 1)/lambda.4 ~ St + Re + Fr +
##     St * Fr + Fr * Re + St * Re, data = train_data)
##
## Coefficients:
## (Intercept)           St           Re           Fr        St:Fr        Re:Fr
##     38.3625      -0.4555      -6.1565     -30.1859       0.2486       5.0834
##       St:Re
##      0.3271
```

Firgure 2.6: Final Linear Models Output

```
lm1 <- lm((R_moment_1^lambda.1 - 1) /
            lambda.1 ~ St + Re + Fr + Fr*Re + St*Fr + Re*St, data = train_data)
summary(lm1)
```

```
##
## Call:
## lm(formula = (R_moment_1^lambda.1 - 1)/lambda.1 ~ St + Re + Fr +
##     Fr * Re + St * Fr + Re * St, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33806 -0.09491  0.00755  0.07987  0.30746
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 17.90588    0.18297   97.861  < 2e-16 ***
## St           0.08917    0.11498    0.776   0.4403
## Re          -4.44003    0.03489 -127.264  < 2e-16 ***
## Fr          -1.78892    0.29634   -6.037 4.38e-08 ***
## Re:Fr        0.34834    0.05546    6.281 1.53e-08 ***
## St:Fr        0.06105    0.02927    2.085   0.0402 *
## St:Re        0.01950    0.02216    0.880   0.3816
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1257 on 82 degrees of freedom
## Multiple R-squared:  0.9976, Adjusted R-squared:  0.9974
## F-statistic:  5641 on 6 and 82 DF,  p-value: < 2.2e-16
```

```r
lm2 <- lm((R_moment_2^lambda.2 - 1) /
            lambda.2 ~ St + Re + Fr + Fr*Re + St*Fr + Re*St, data = train_data)
summary(lm2)
```

```
##
## Call:
## lm(formula = (R_moment_2^lambda.2 - 1)/lambda.2 ~ St + Re + Fr +
##     Fr * Re + St * Fr + Re * St, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7761 -0.9047  0.2701  1.3523  2.8961
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.2875     2.8219  11.796  < 2e-16 ***
## St           -0.5535     1.7733  -0.312 0.755749
## Re           -6.5838     0.5381 -12.236  < 2e-16 ***
## Fr          -18.5954     4.5704  -4.069 0.000108 ***
## Re:Fr         3.1739     0.8554   3.711 0.000375 ***
## St:Fr         0.1939     0.4515   0.429 0.668745
## St:Re         0.3174     0.3418   0.929 0.355812
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.939 on 82 degrees of freedom
## Multiple R-squared:  0.7857, Adjusted R-squared:   0.77
## F-statistic: 50.11 on 6 and 82 DF,  p-value: < 2.2e-16
```

```r
lm3 <- lm((R_moment_3^lambda.3 - 1) /
            lambda.3 ~ St + Re + Fr + Fr*Re + St*Fr + Re*St, data = train_data)
summary(lm3)
```

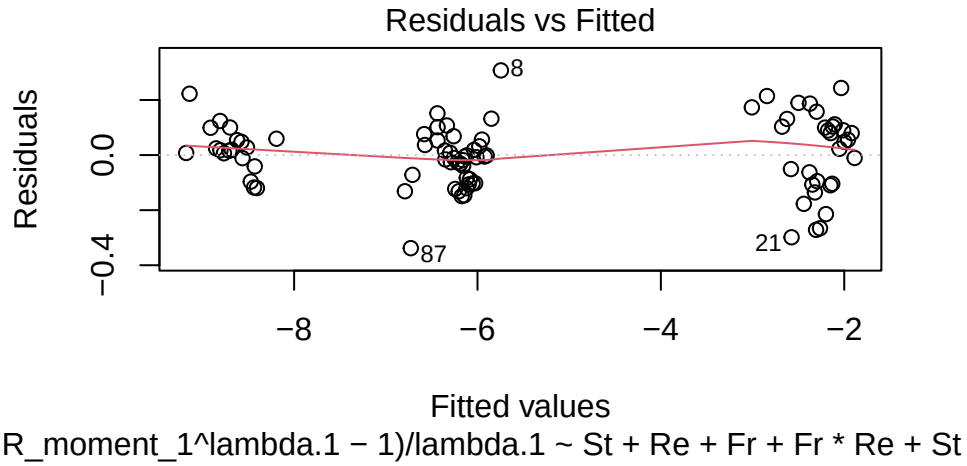
```
##
## Call:
## lm(formula = (R_moment_3^lambda.3 - 1)/lambda.3 ~ St + Re + Fr +
##     Fr * Re + St * Fr + Re * St, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0882  -1.2739   0.4821   1.7402   3.8787
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.8073     3.7702  10.028 6.58e-16 ***
## St           -0.0296     2.3692  -0.012    0.990
## Re           -6.6125     0.7189  -9.198 2.90e-14 ***
## Fr          -28.1339     6.1062  -4.607 1.48e-05 ***
## Re:Fr         4.7601     1.1428   4.165 7.65e-05 ***
## St:Fr         0.1398     0.6032   0.232    0.817
## St:Re         0.2356     0.4566   0.516    0.607
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.59 on 82 degrees of freedom
## Multiple R-squared:  0.6699, Adjusted R-squared:  0.6458
## F-statistic: 27.74 on 6 and 82 DF,  p-value: < 2.2e-16
```

```
lm4 <- lm((R_moment_4^lambda.4 - 1) /
            lambda.4 ~ St + Re + Fr + Fr*Re + St*Fr + Re*St, data = train_data)
summary(lm4)
```

```
##
## Call:
## lm(formula = (R_moment_4^lambda.4 - 1)/lambda.4 ~ St + Re + Fr +
##     Fr * Re + St * Fr + Re * St, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.1287  -1.4078   0.6943   2.0033   4.1905
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  38.3625     4.3420   8.835 1.53e-13 ***
## St           -0.4555     2.7285  -0.167 0.867839
## Re           -6.1565     0.8279  -7.436 9.03e-11 ***
## Fr          -30.1859     7.0322  -4.293 4.81e-05 ***
## Re:Fr         5.0834     1.3161   3.862 0.000223 ***
## St:Fr         0.2486     0.6947   0.358 0.721341
## St:Re         0.3271     0.5259   0.622 0.535699
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.983 on 82 degrees of freedom
## Multiple R-squared:  0.5943, Adjusted R-squared:  0.5647
## F-statistic: 20.02 on 6 and 82 DF,  p-value: 2.833e-14
```

Figure 2.7-8: Diagnostic Plots For Final Linear Model of R Moment 1



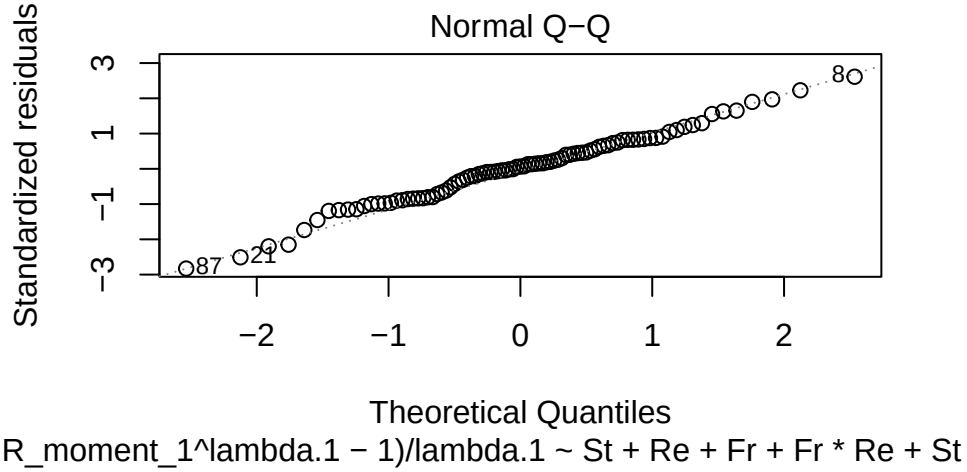
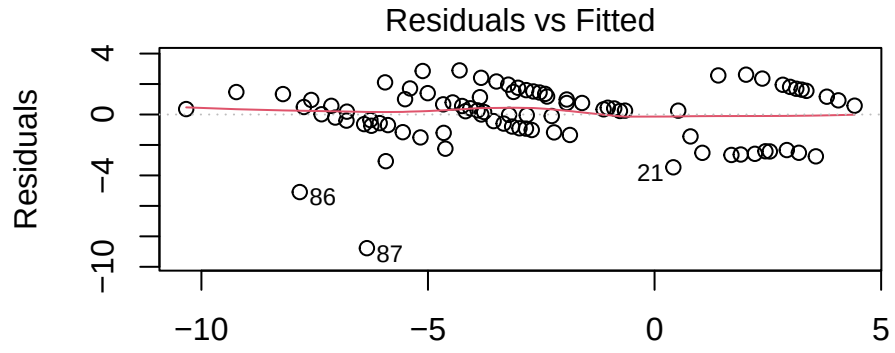


Figure 2.9-10: Diagnostic Plots For Final Linear Model of R Moment 2



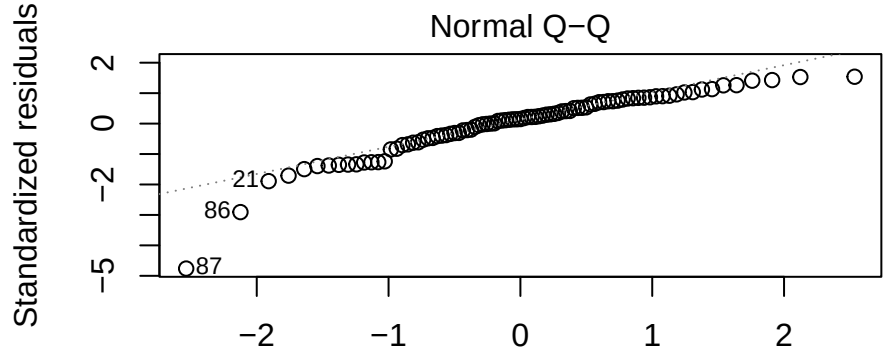
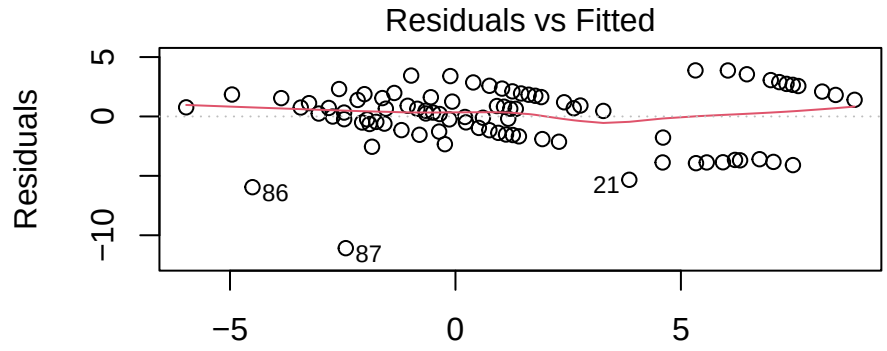


Figure 2.11-12: Diagnostic Plots For Final Linear Model of R Moment 3



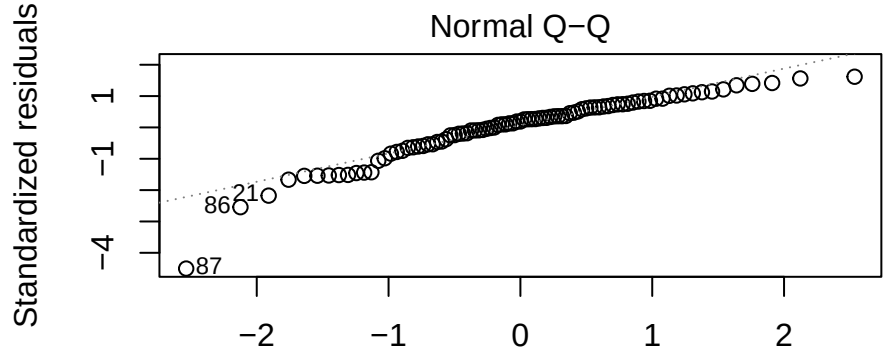
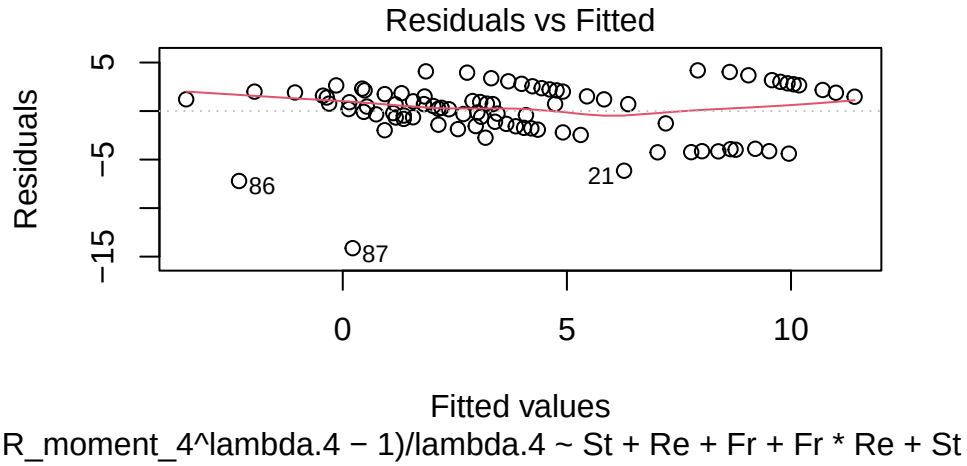


Figure 2.13-14: Diagnostic Plots For Final Linear Model of R Moment 4



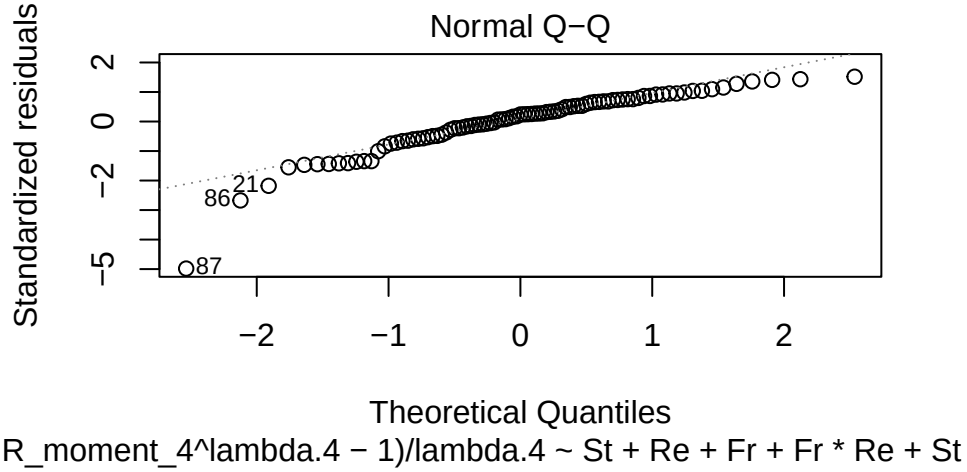


Figure 3.1: 10-fold CV estimated Test Error for GAM Models of Each Moment

```
## # A tibble: 4 x 8
##    Model.0  Model.1  Model.2  Model.3  Model.4  Model.5  Model.6  Model.7
##      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 1.34e- 4 1.36e- 4 1.56e- 4 1.04e- 4 1.09e- 4 7.12e- 5 1.59e- 4 1.09e- 4
## 2 7.20e+ 4 7.23e+ 4 7.20e+ 4 6.31e+ 4 6.52e+ 4 6.28e+ 4 7.22e+ 4 6.52e+ 4
## 3 5.22e+12 5.22e+12 5.22e+12 5.18e+12 5.19e+12 5.17e+12 5.22e+12 5.19e+12
## 4 3.65e+20 3.65e+20 3.65e+20 3.65e+20 3.65e+20 3.65e+20 3.65e+20 3.65e+20
```