Case Study

Abdel Shehata

2022-10-26

Introduction and Data

Introduction

Data Introduction

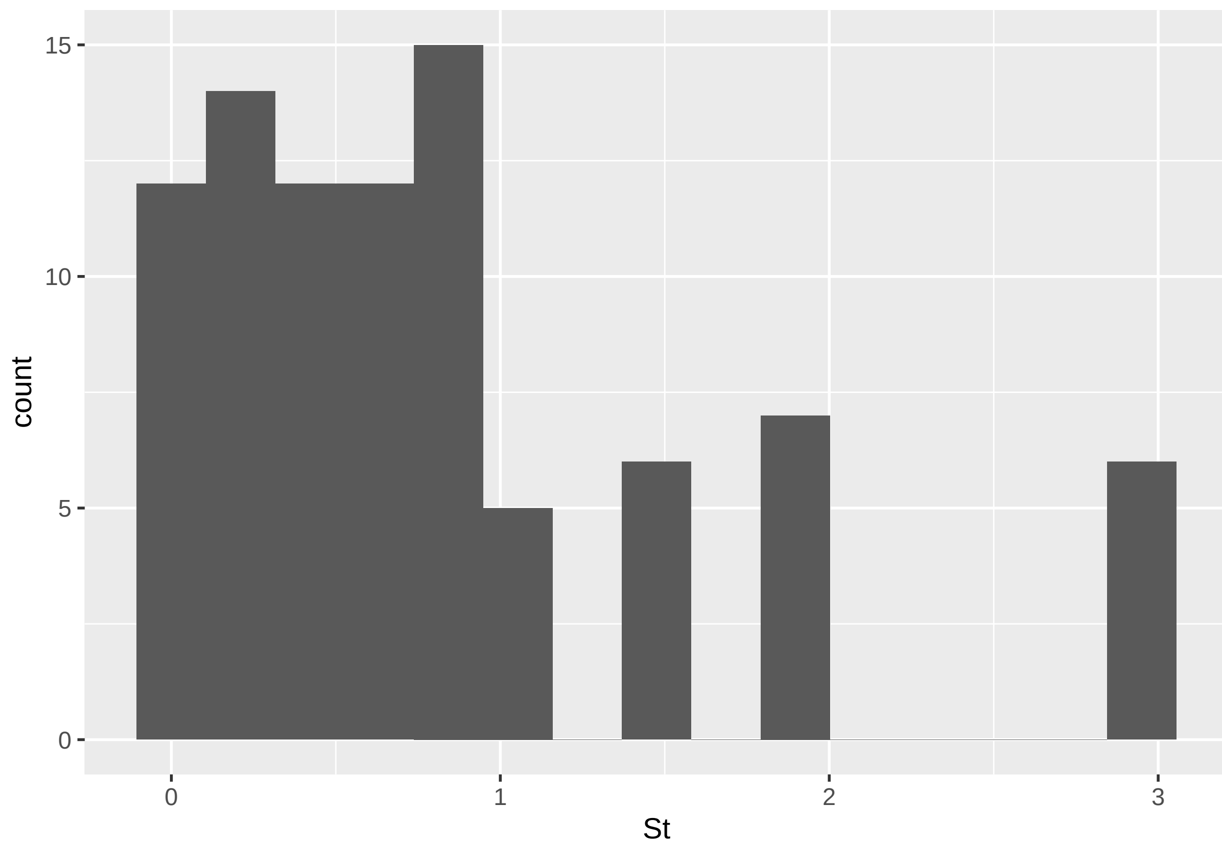
Exploratory Data Analysis

```
library(readr)
data_train <- read_csv("data-train.csv")
```

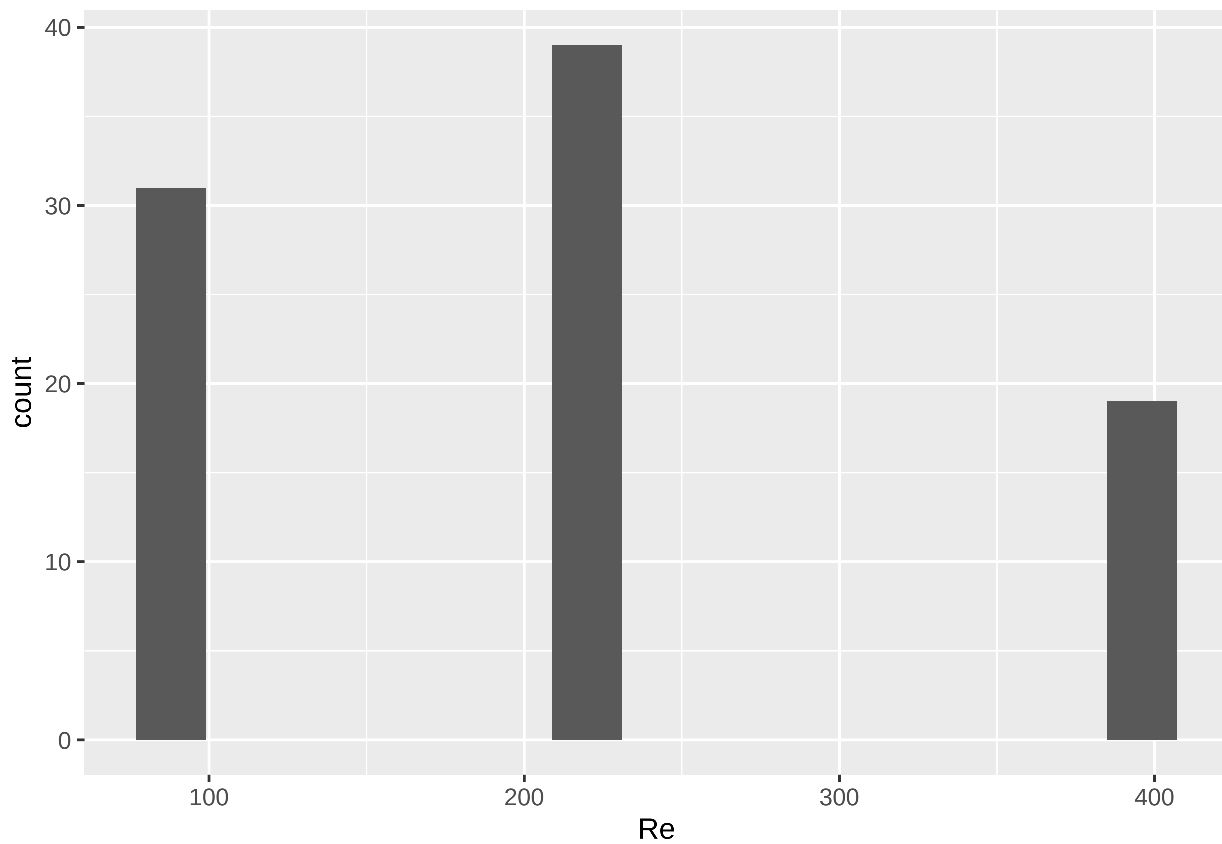
```
## Rows: 89 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (7): St, Re, Fr, R_moment_1, R_moment_2, R_moment_3, R_moment_4
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
attach(data_train)
data_train <- data_train%>% mutate(TFr = case_when(Fr>1~ .99999, Fr<1~Fr))
data_train<-data_train%>%mutate(TFr=logit(TFr))

ggplot(data_train) +
  geom_histogram(aes(x = St), bins = 15)
```

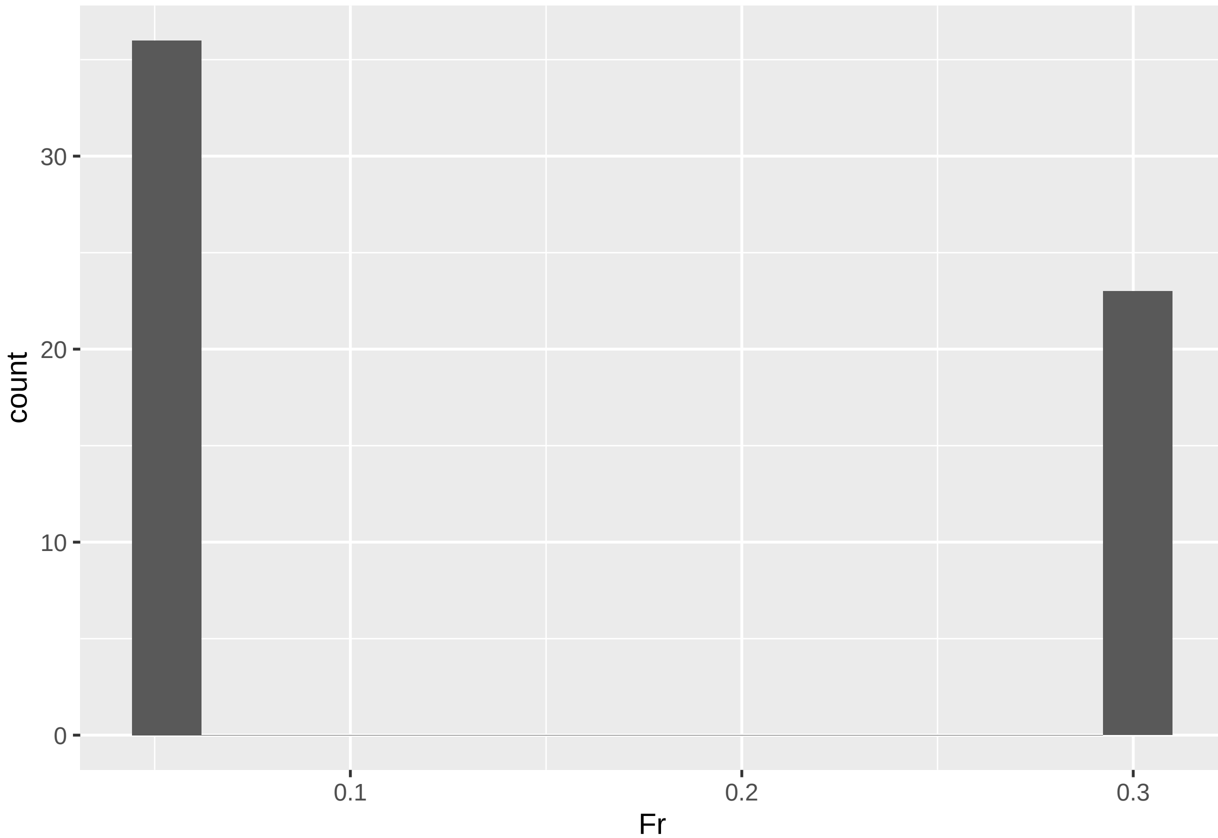



```r
ggplot(data_train) +
  geom_histogram(aes(x = Re), bins = 15)
```

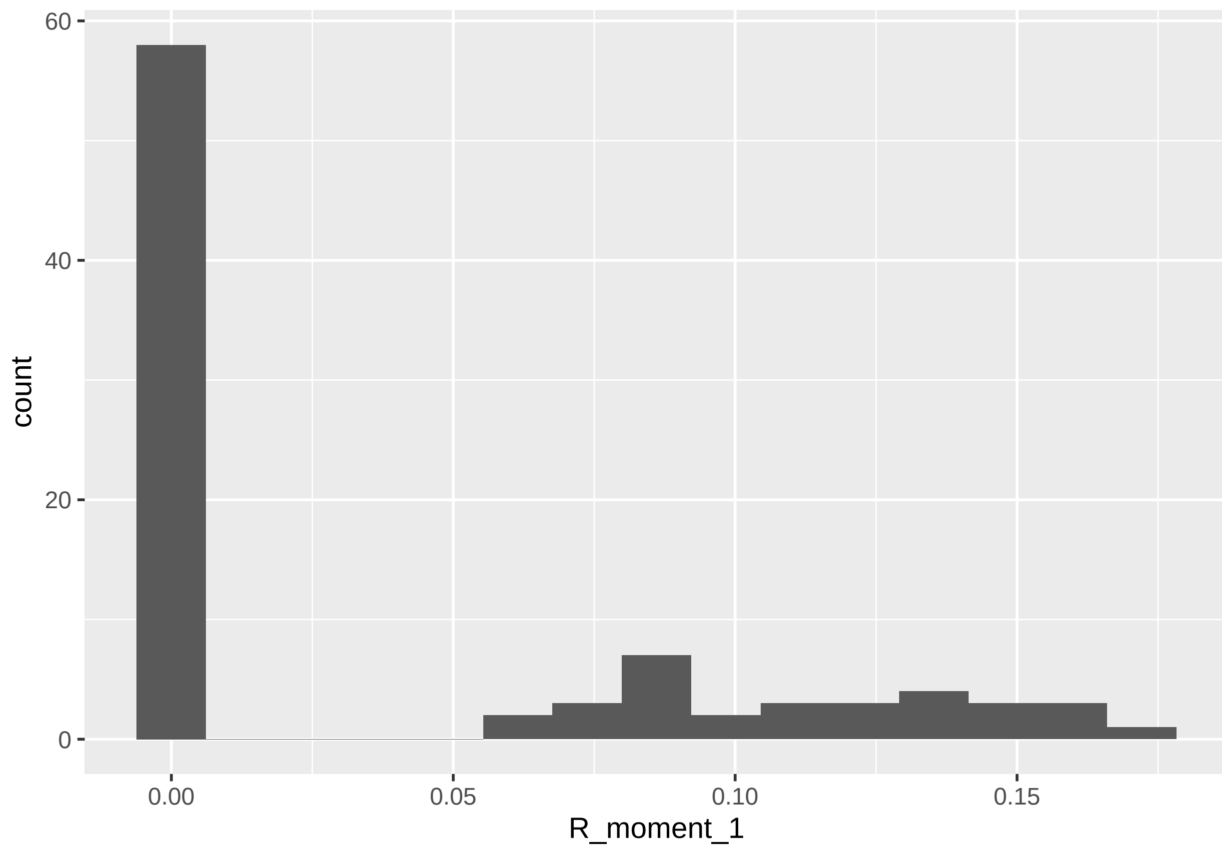



```r
ggplot(data_train) +
  geom_histogram(aes(x = Fr), bins = 15)
```

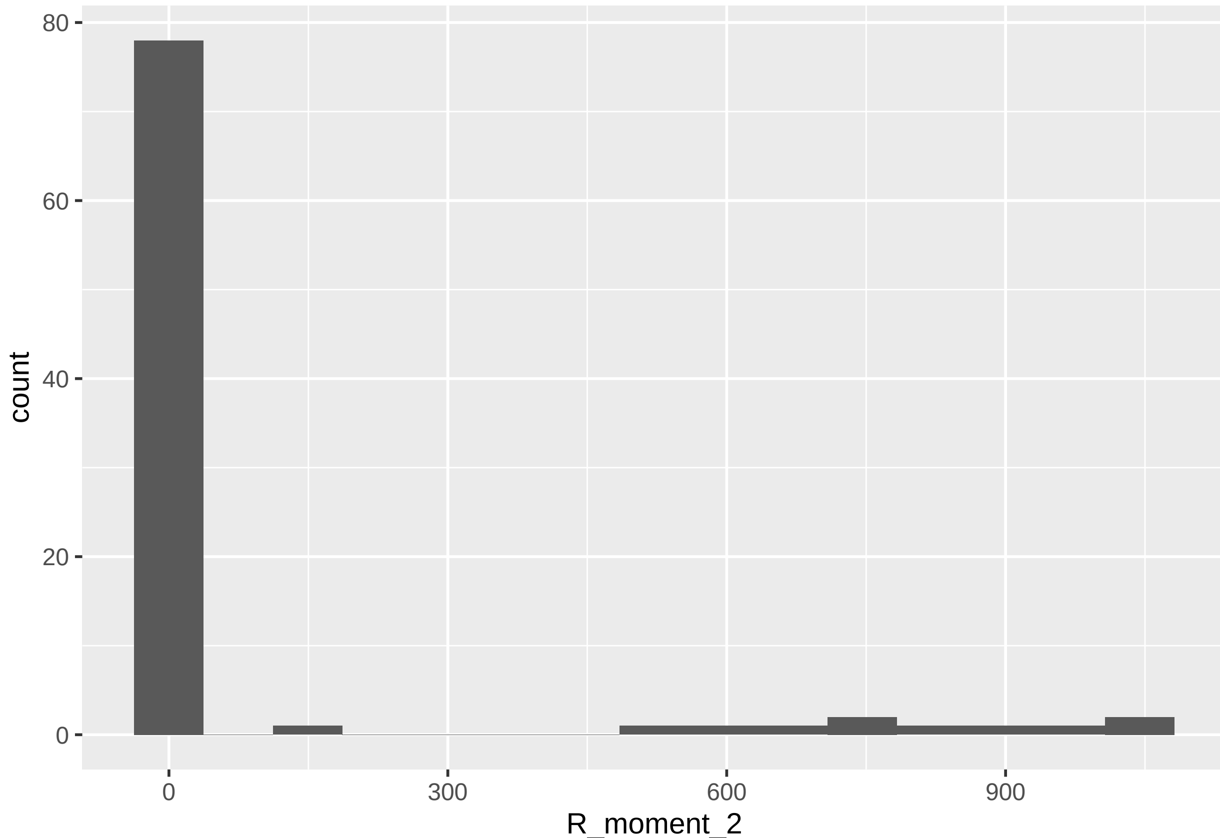
```
## Warning: Removed 30 rows containing non-finite values (stat_bin).
```



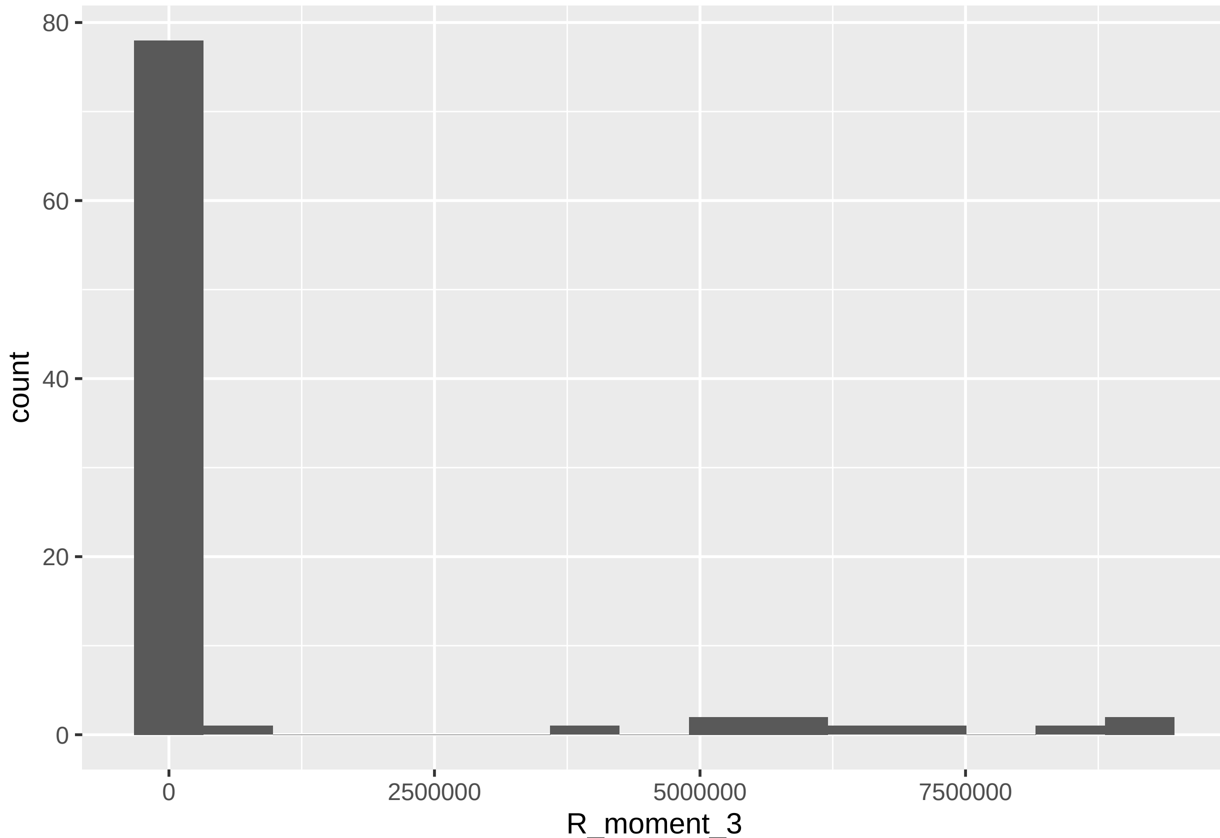

```
ggplot(data_train) +
  geom_histogram(aes(x = R_moment_1), bins = 15)
```



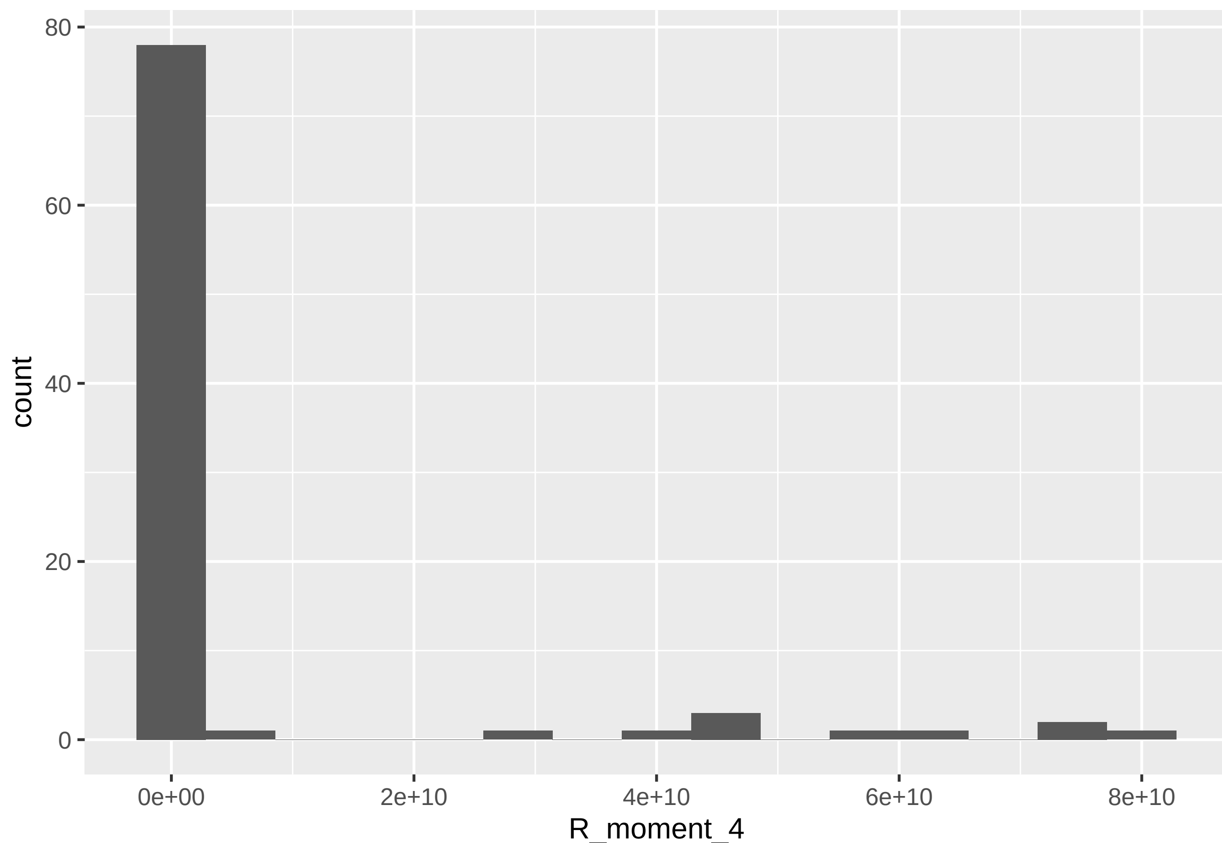

```
ggplot(data_train) +
  geom_histogram(aes(x = R_moment_2), bins = 15)
```




```
ggplot(data_train) +
  geom_histogram(aes(x = R_moment_3), bins = 15)
```




```
ggplot(data_train) +
  geom_histogram(aes(x = R_moment_4), bins = 15)
```



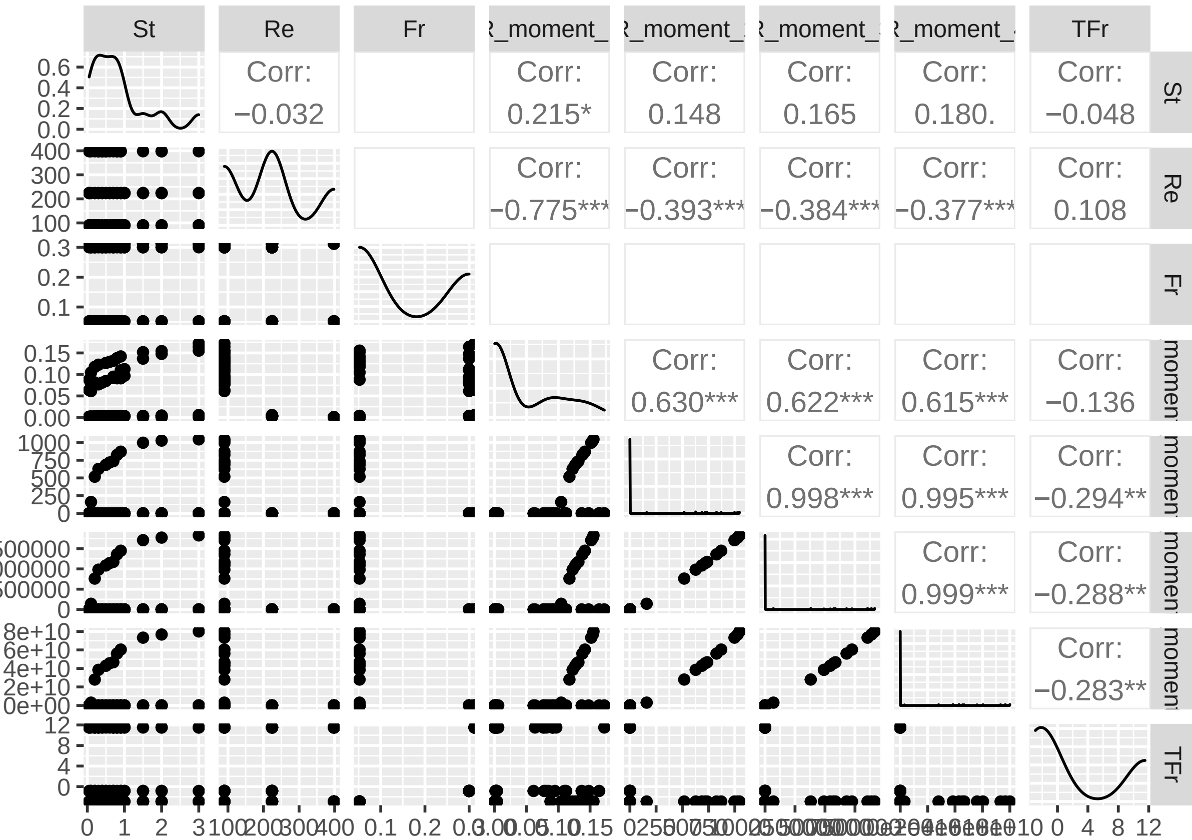

```r
ggpairs(data_train)
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```

```
## Warning: Removed 30 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
## Removed 1 rows containing missing values (geom_text).
## Removed 1 rows containing missing values (geom_text).
## Removed 1 rows containing missing values (geom_text).
## Removed 1 rows containing missing values (geom_text).
```



Some brief notes:

Observations on predictors: St (size) seems to be mostly small particles with some trials with larger particles. Re (turbulence) seems to be in three groups: low (90), medium (224), and high (398). Perhaps it could be considered a categorical variable? Fr (gravitational acceleration) seems to also be in three groups: low (.052), medium (.3), and high (infinite). Could this also become a categorical variable? We have decided to do a logistic transformation on Fr in order to approximate the effects of infinity.

Also, I believe we should centralize the second through fourth moments. The first raw moment is actually helpful because it tells us about the average amount of turbulence. However, when it comes to the shape of the distribution (variance, skewness, and kurtosis) we need to centralize the moments in order to interpret them.

The code for transforming the variables is below:

```
data_train <- data_train %>% mutate(R_moment_1_central = 0)
data_train <- data_train %>% mutate(R_moment_2_central = R_moment_2 - (R_moment_1)^2)
data_train <- data_train %>% mutate(R_moment_3_central = R_moment_3 - 3*R_moment_1*R_moment_2 + 2*(R_mo
data_train <- data_train %>% mutate(R_moment_4_central = R_moment_4 - 4*R_moment_1*R_moment_3 + 6*((R_m
```

Correlations: Reynolds number is negatively correlated with all moments, which is surprising but I believe it is due to the fact that almost all of the observations of all the moments are mostly around 0 with only some exceptions. The 2nd, 3rd, and 4th moments are pretty correlated but this makes sense because they are all various measures of the width and shape of the tails.

Plots: St and the first moment seem to have a linear or quadratic relationship. I would not be surprised if it is true that bigger particles cluster more on average. St and the second, third, and fourth moments seem to have a linear or quadratic relationship. Perhaps bigger particles behave more unpredictably.

Methodology

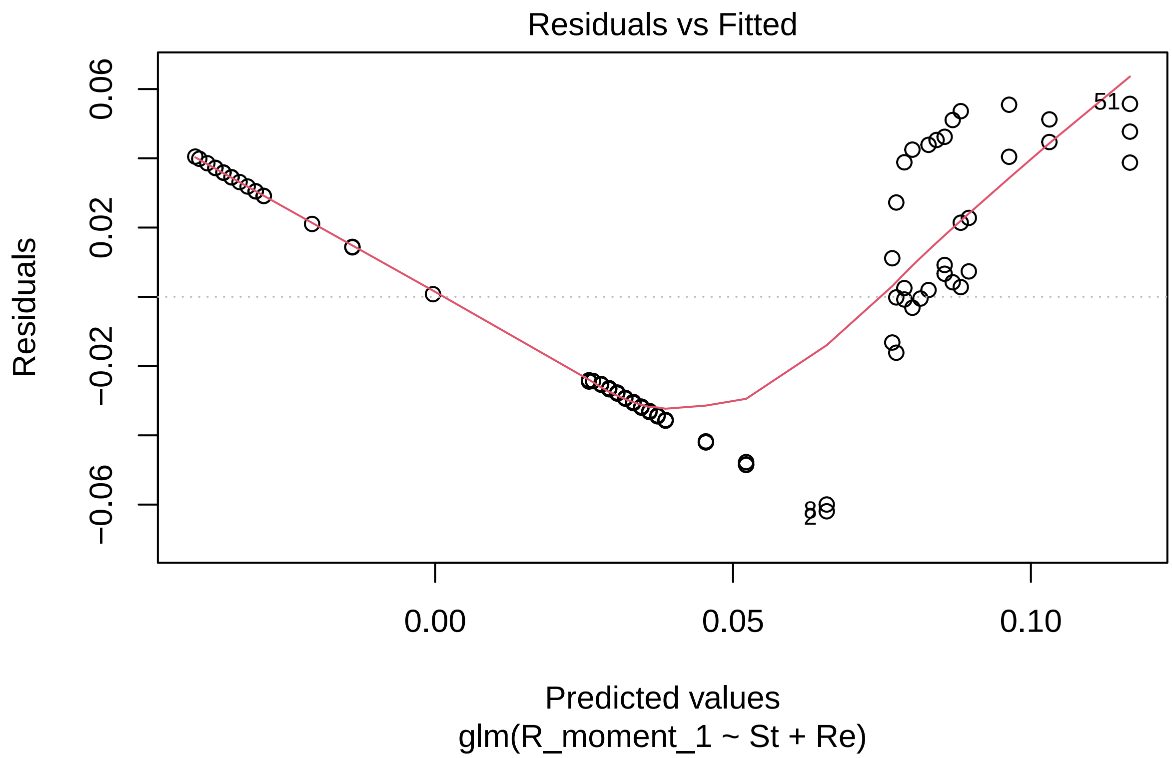
Linear

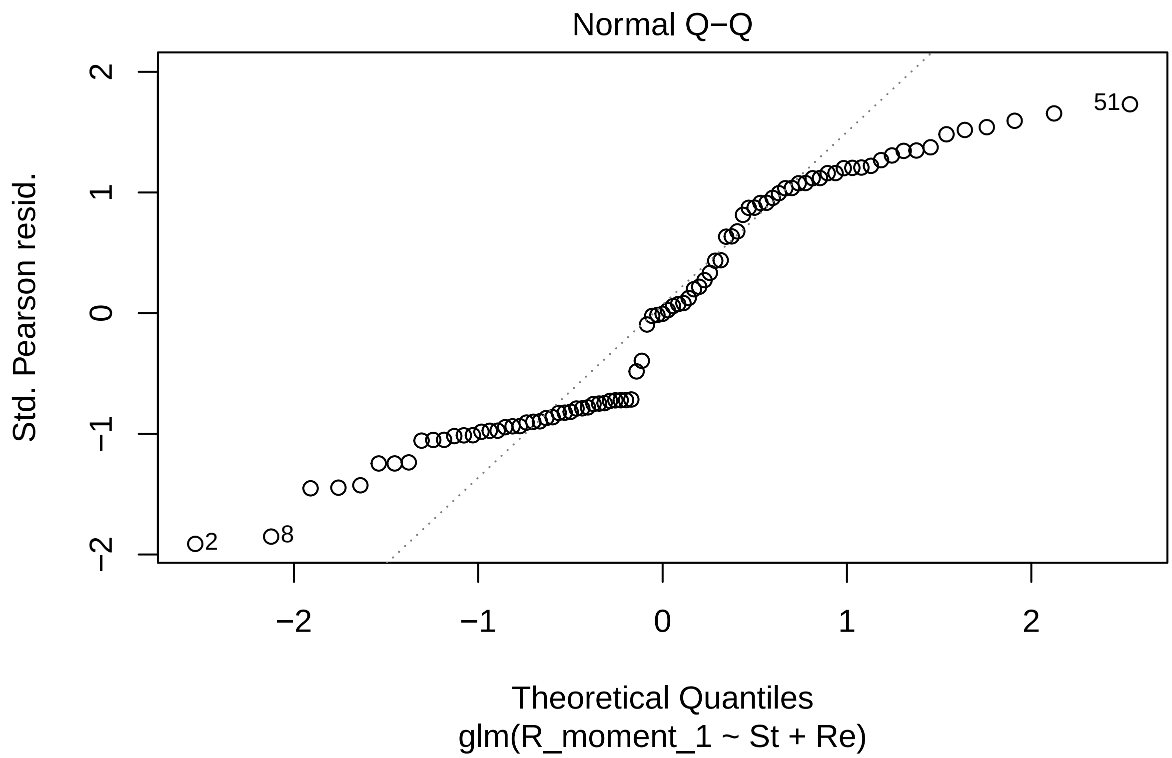
```
full_linear_E1 <- glm(R_moment_1 ~ St + TFr + Re, data = data_train)
step_full_linear_E1 <- stepAIC(full_linear_E1, direction = "both", trace = FALSE)
summary(step_full_linear_E1)
```

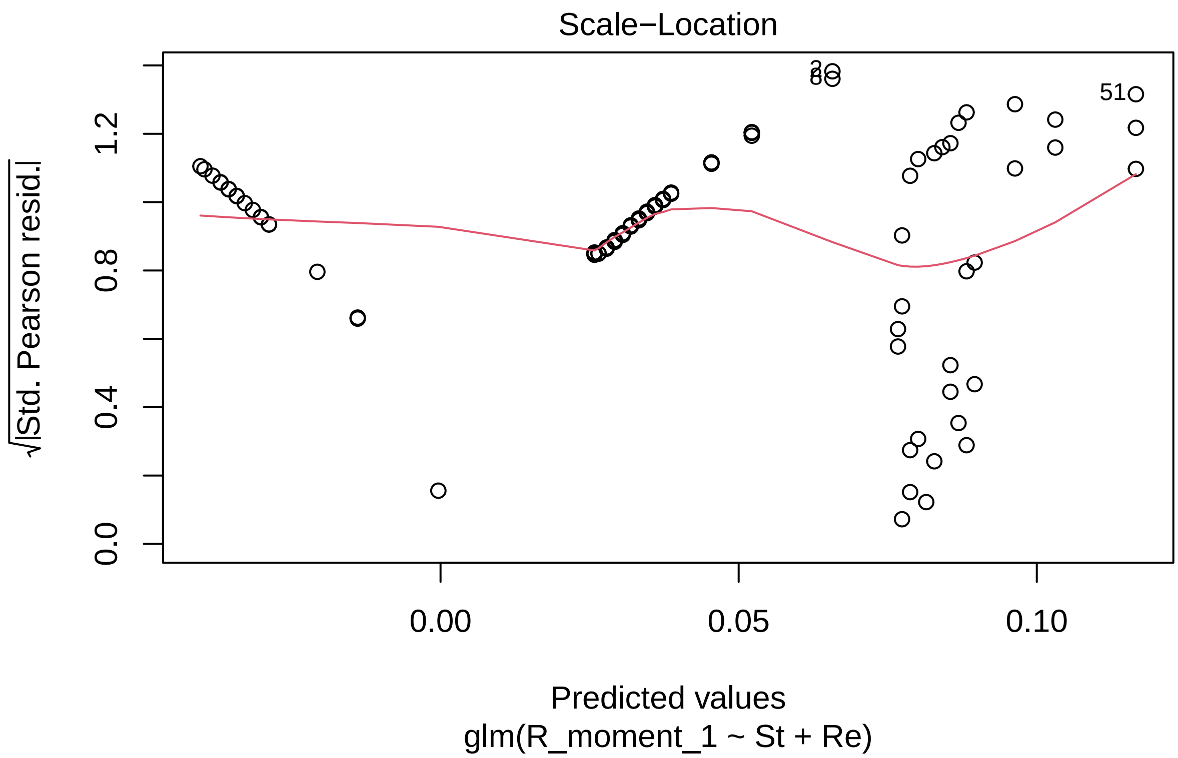
Linear Fitting

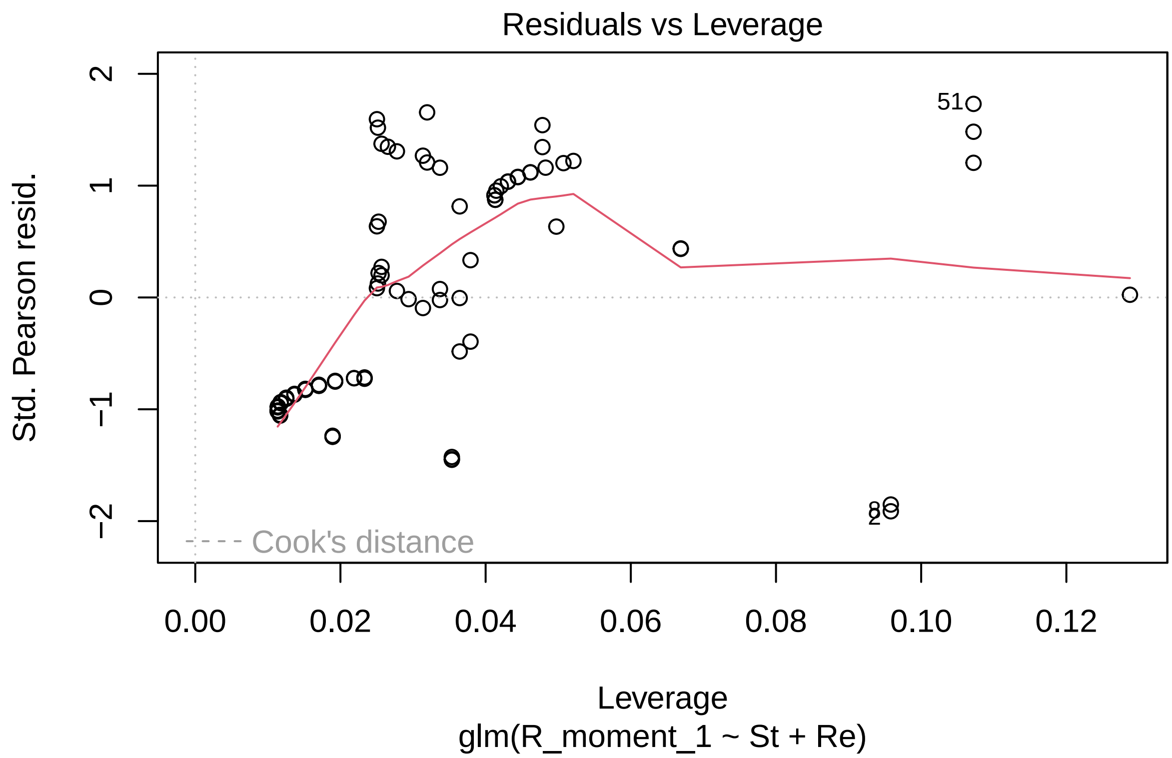
```
##
## Call:
## glm(formula = R_moment_1 ~ St + Re, data = data_train)
##
## Deviance Residuals:
##       Min         1Q     Median         3Q        Max
## -0.061936  -0.030347  -0.000174   0.034491   0.055714
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.102e-01  8.837e-03  12.475  < 2e-16 ***
## St           1.353e-02  4.621e-03   2.927  0.00438 **
## Re          -3.798e-04  3.215e-05 -11.816  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.001160225)
##
##     Null deviance: 0.274427  on 88  degrees of freedom
## Residual deviance: 0.099779  on 86  degrees of freedom
## AIC: -344.04
##
## Number of Fisher Scoring iterations: 2
```

```
plot(step_full_linear_E1)
```


Residuals vs Fitted
Residuals
0.06
0.02
−0.02
−0.06
0.00
0.05
0.10
51
8
2
Predicted values
glm(R_moment_1 ~ St + Re)


Normal Q−Q
Std. Pearson resid.
Theoretical Quantiles
glm(R_moment_1 ~ St + Re)
2
8
51
−2
−1
0
1
2


Scale−Location
√|Std. Pearson resid.|
0.0
0.4
0.8
1.2
0.00
0.05
0.10
Predicted values
glm(R_moment_1 ~ St + Re)
51

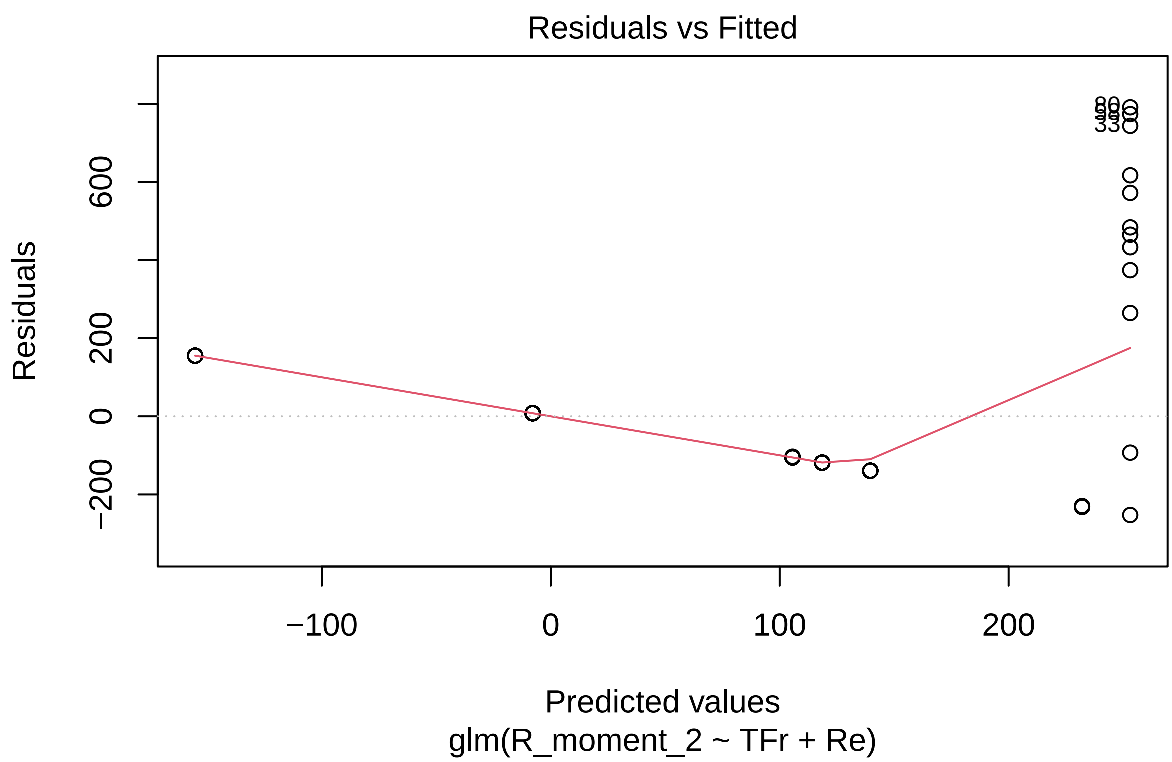


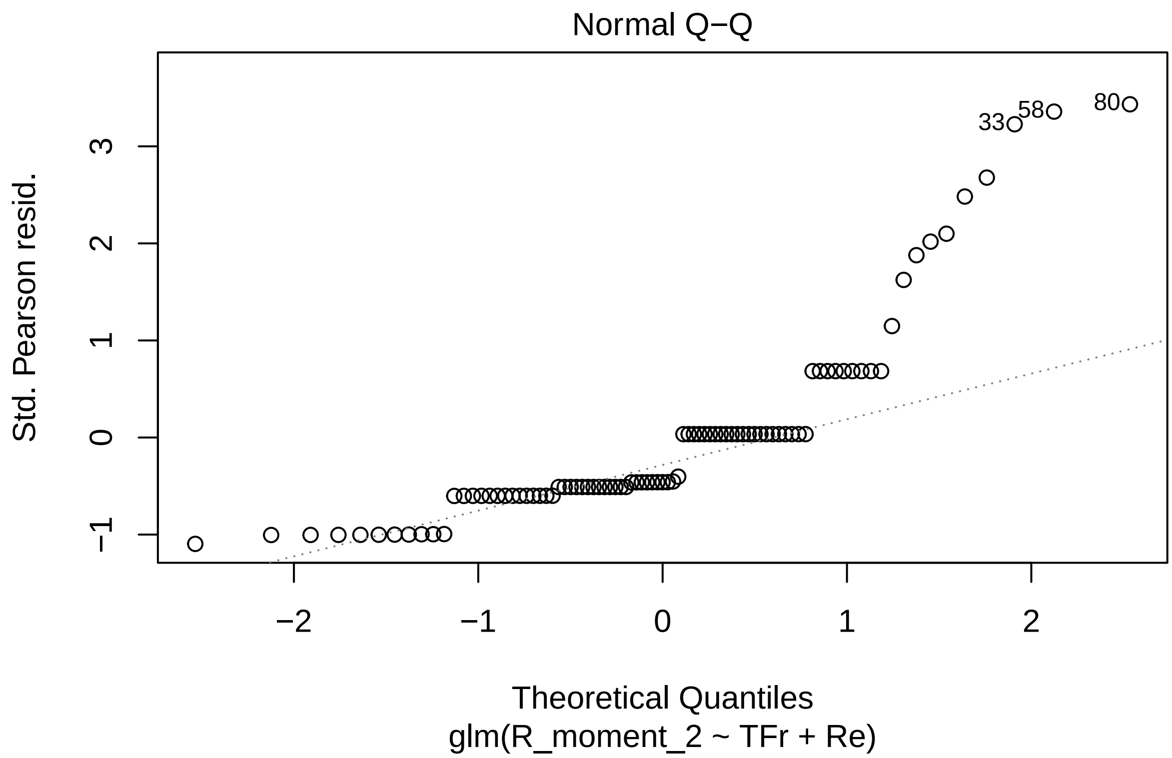

```
full_linear_E2 <- glm(R_moment_2 ~ St + TFr + Re, data = data_train)
step_full_linear_E2 <- stepAIC(full_linear_E2, direction = "both", trace = FALSE)
summary(step_full_linear_E2)
```

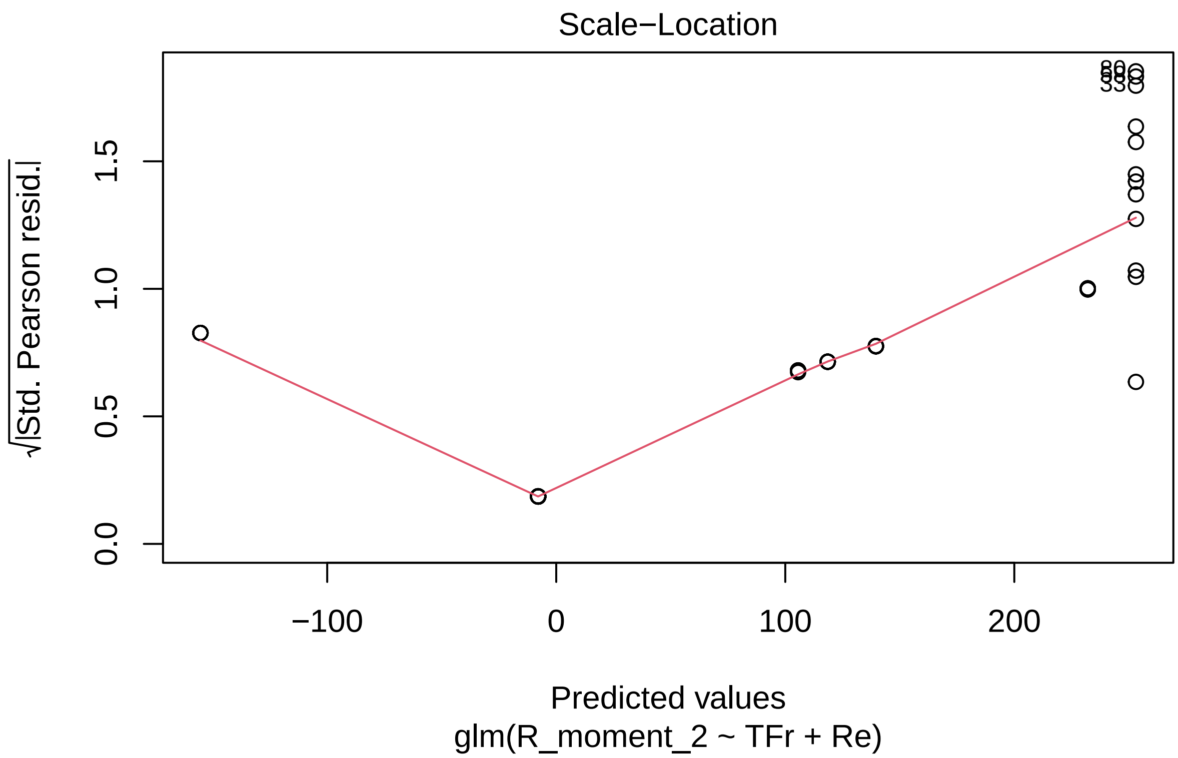
```
##
## Call:
## glm(formula = R_moment_2 ~ TFr + Re, data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -252.57  -139.17  -104.99     7.97   791.19
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 299.6593    53.6457   5.586 2.67e-07 ***
## TFr         -10.2317     3.8471  -2.660 0.009332 **
## Re           -0.8473     0.2221  -3.815 0.000256 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 54790.94)
##
##     Null deviance: 6032373  on 88  degrees of freedom
## Residual deviance: 4712021  on 86  degrees of freedom
## AIC: 1228.6
```

```
##
## Number of Fisher Scoring iterations: 2
```

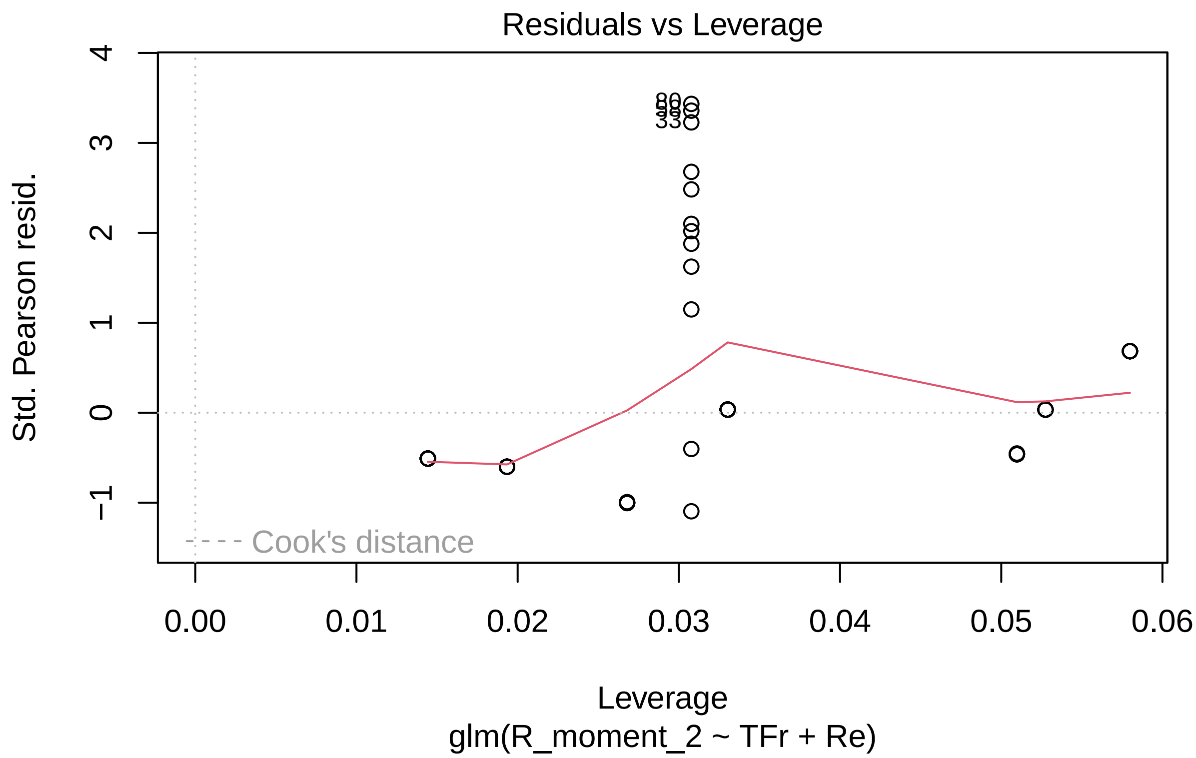
```
plot(step_full_linear_E2)
```




Normal Q−Q
Std. Pearson resid.
Theoretical Quantiles
glm(R_moment_2 ~ TFr + Re)
33
58
80
−1
0
1
2
3
−2
−1
0
1
2



Scale−Location
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
−100
0
100
200
Predicted values
glm(R_moment_2 ~ TFr + Re)

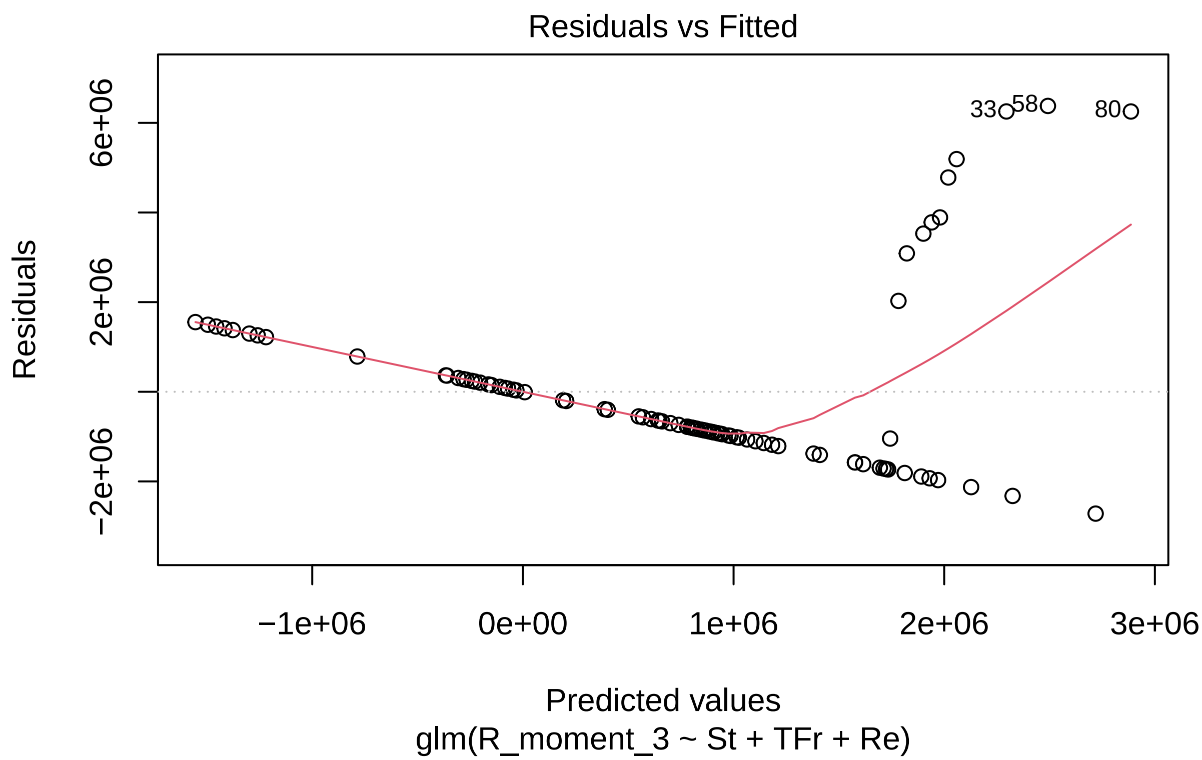


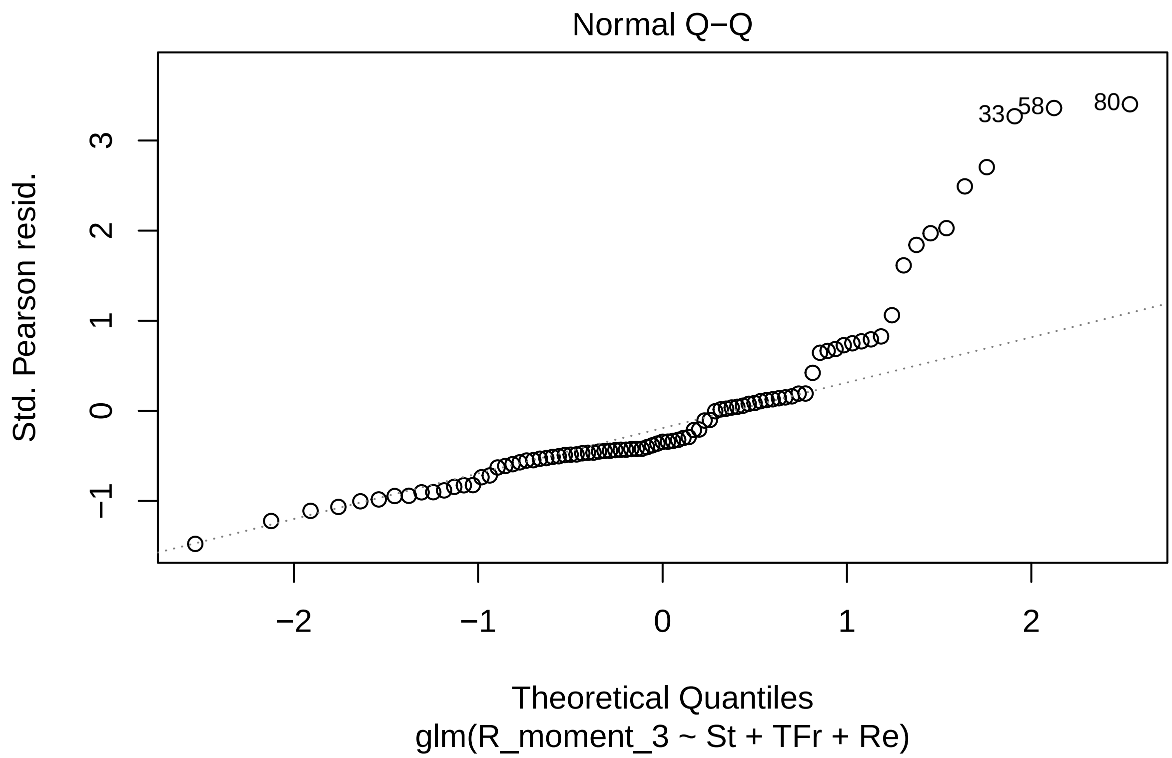

```r
full_linear_E3 <- glm(R_moment_3 ~ St + TFr + Re, data = data_train)
step_full_linear_E3 <- stepAIC(full_linear_E3, direction = "both", trace = FALSE)
summary(step_full_linear_E3)
```

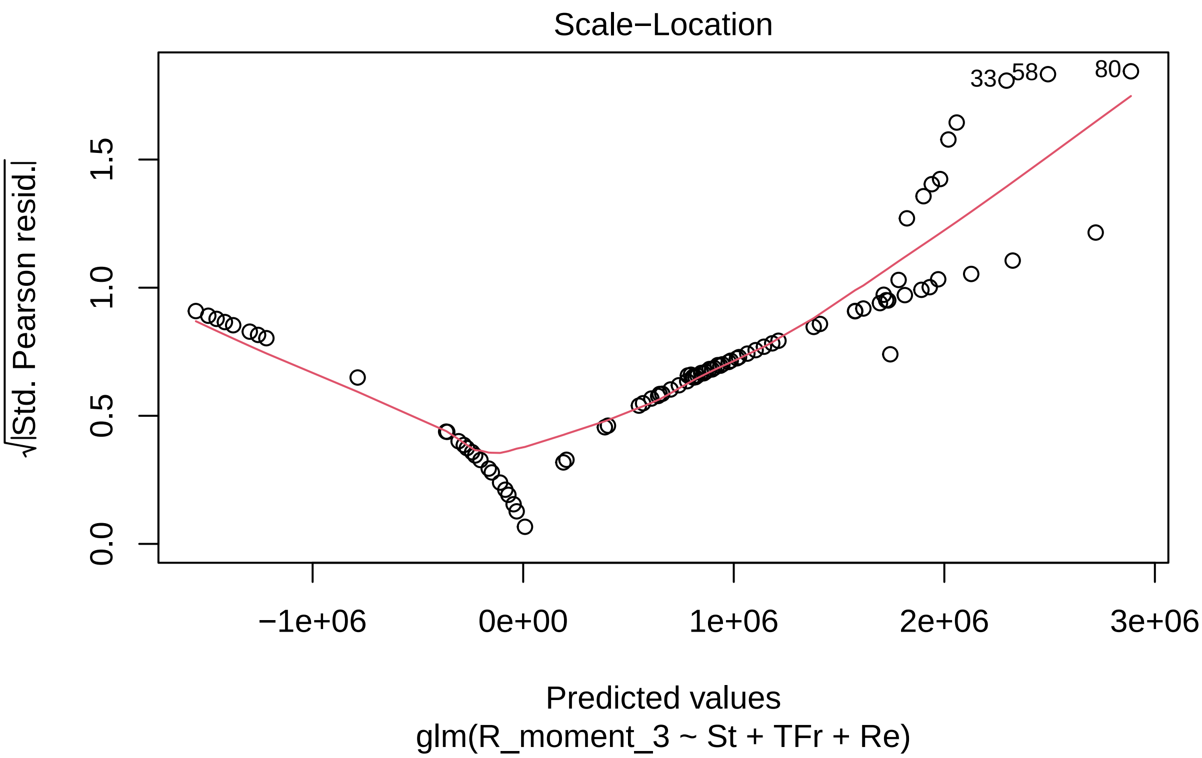
```
##
## Call:
## glm(formula = R_moment_3 ~ St + TFr + Re, data = data_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2718782  -1025004   -660570    281888   6377805
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2082451     508059   4.099 9.46e-05 ***
## St            394073     264795   1.488 0.140394
## TFr           -81448      32077  -2.539 0.012933 *
## Re             -6832       1851  -3.691 0.000393 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 3.801448e+12)
##
##     Null deviance: 4.1938e+14  on 88  degrees of freedom
## Residual deviance: 3.2312e+14  on 85  degrees of freedom
```

```
## AIC: 2836.5
##
## Number of Fisher Scoring iterations: 2
```

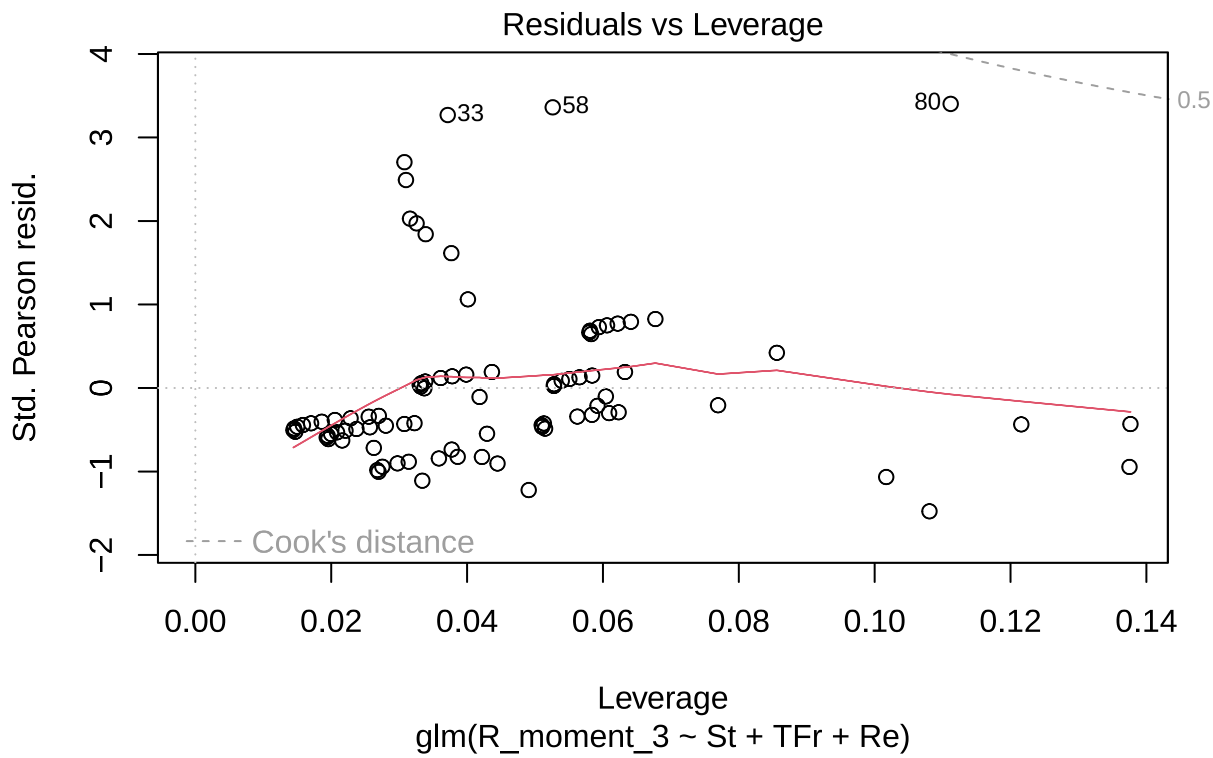
```
plot(step_full_linear_E3)
```




Normal Q−Q
Std. Pearson resid.
3
2
1
0
−1
33
58
80
−2
−1
0
1
2
Theoretical Quantiles
glm(R_moment_3 ~ St + TFr + Re)



Scale−Location
33
58
80
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
−1e+06
0e+00
1e+06
2e+06
3e+06
Predicted values
glm(R_moment_3 ~ St + TFr + Re)

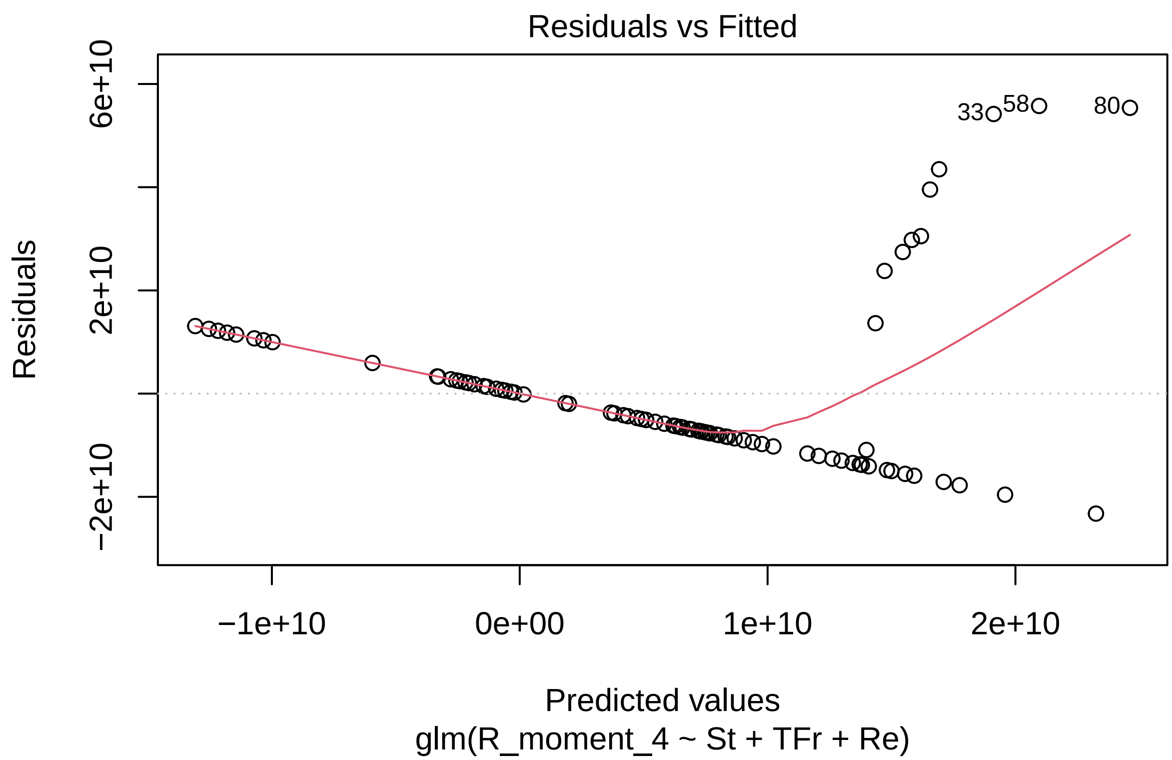


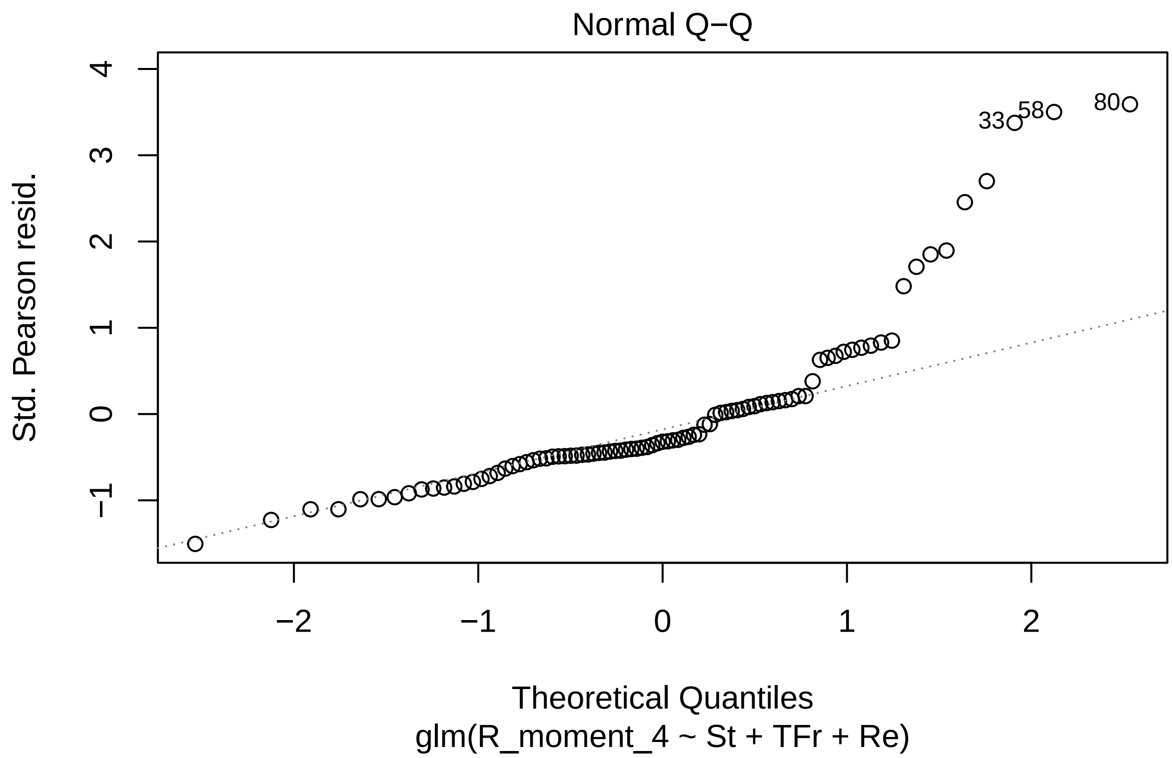

```
full_linear_E4 <- glm(R_moment_4 ~ St + TFr + Re, data = data_train)
step_full_linear_E4 <- stepAIC(full_linear_E4, direction = "both", trace = FALSE)
summary(step_full_linear_E4)
```

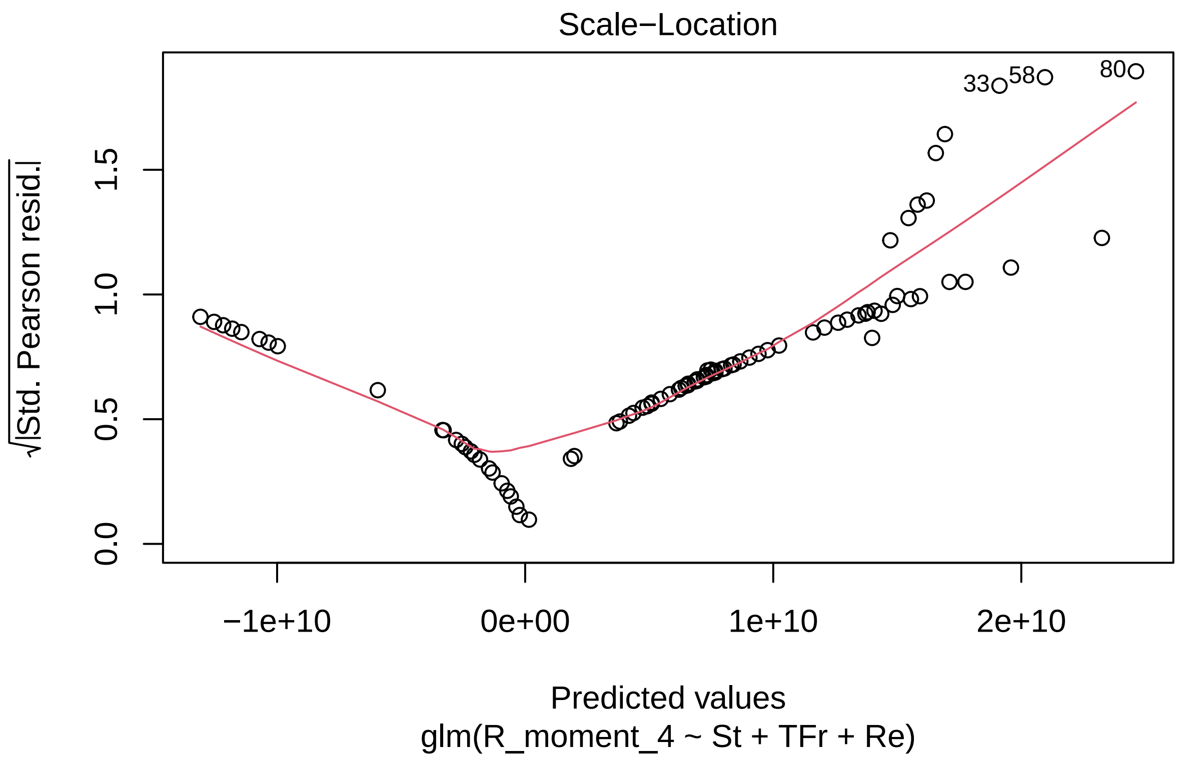
```
##
## Call:
## glm(formula = R_moment_4 ~ St + TFr + Re, data = data_train)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.325e+10  -8.400e+09  -5.101e+09   2.554e+09   5.575e+10
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.673e+10  4.263e+09   3.925 0.000175 ***
## St           3.667e+09  2.222e+09   1.651 0.102520
## TFr         -6.673e+08  2.691e+08  -2.480 0.015126 *
## Re          -5.609e+07  1.553e+07  -3.612 0.000513 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2.676008e+20)
##
##     Null deviance: 2.9413e+22  on 88  degrees of freedom
## Residual deviance: 2.2746e+22  on 85  degrees of freedom
```

```
## AIC: 4444.7
##
## Number of Fisher Scoring iterations: 2
```

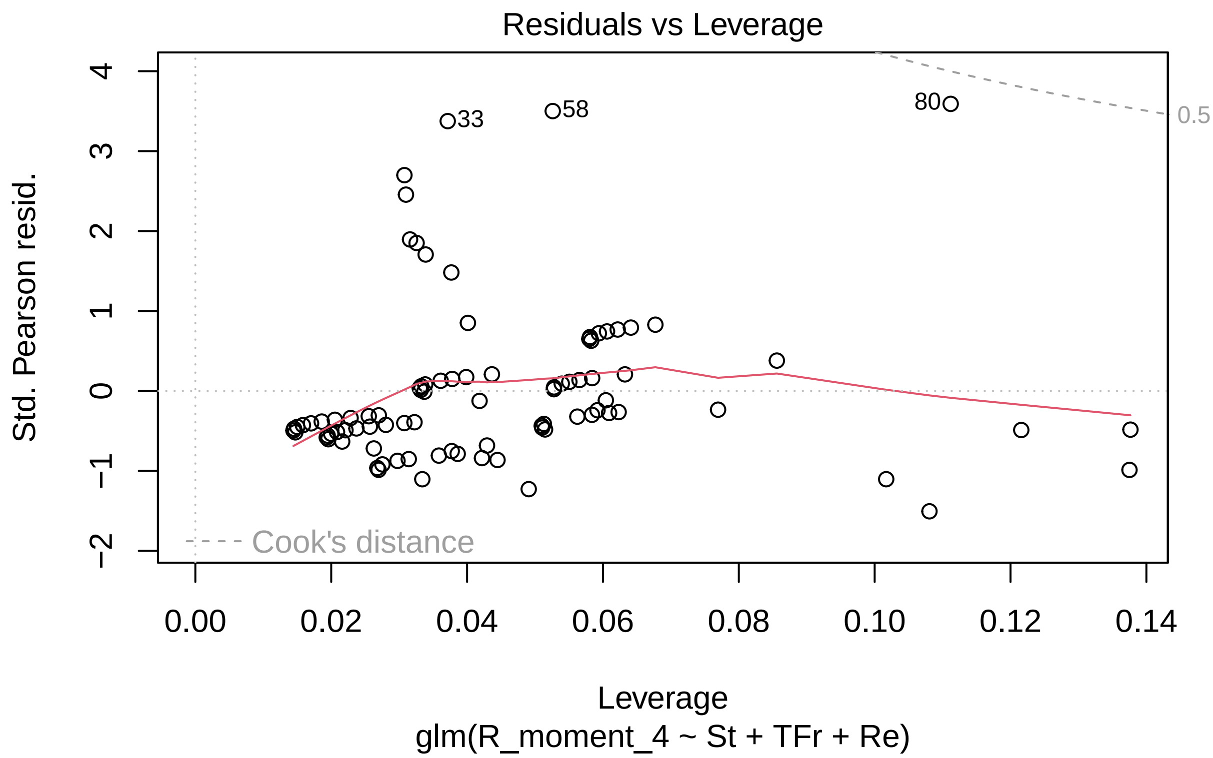
```
plot(step_full_linear_E4)
```




Normal Q−Q
Std. Pearson resid.
Theoretical Quantiles
glm(R_moment_4 ~ St + TFr + Re)
33
58
80
−2
−1
0
1
2
3
4



Scale−Location
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
33
58
80
−1e+10
0e+00
1e+10
2e+10
Predicted values
glm(R_moment_4 ~ St + TFr + Re)



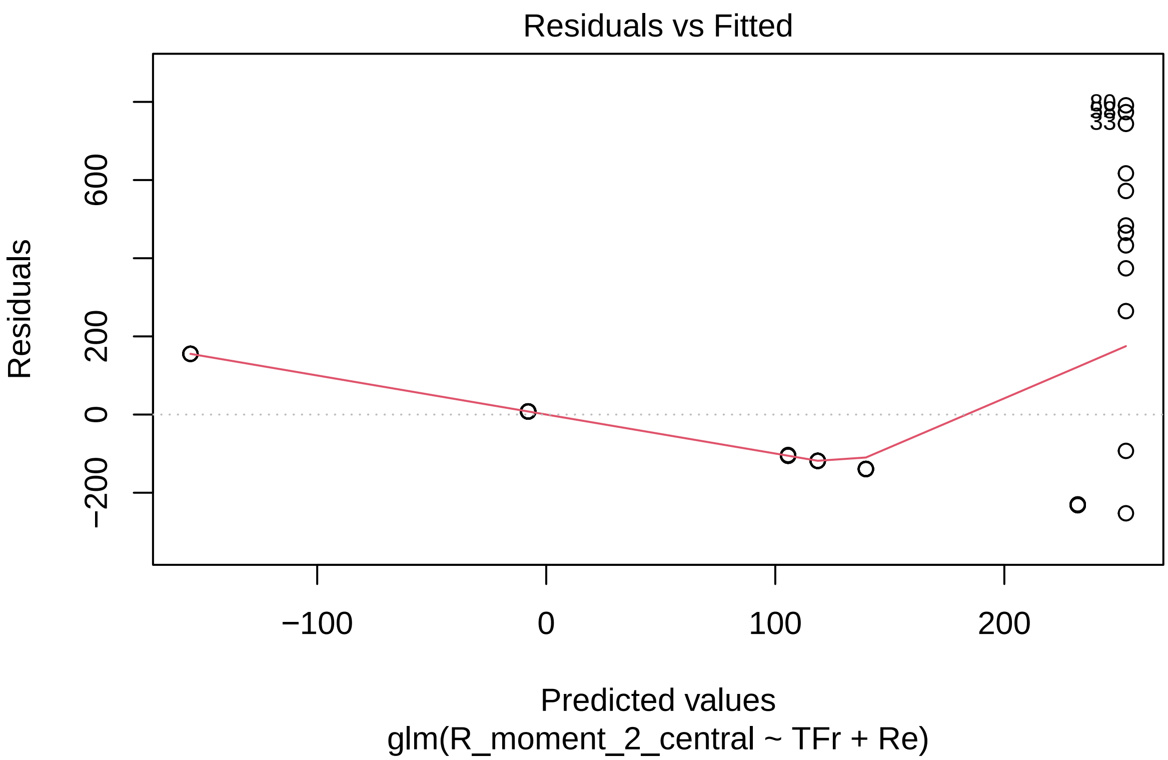

```
full_linear_E2_central <- glm(R_moment_2_central ~ St + TFr + Re, data = data_train)
step_full_linear_E2_central <- stepAIC(full_linear_E2_central, direction = "both", trace = FALSE)
summary(step_full_linear_E2_central)
```

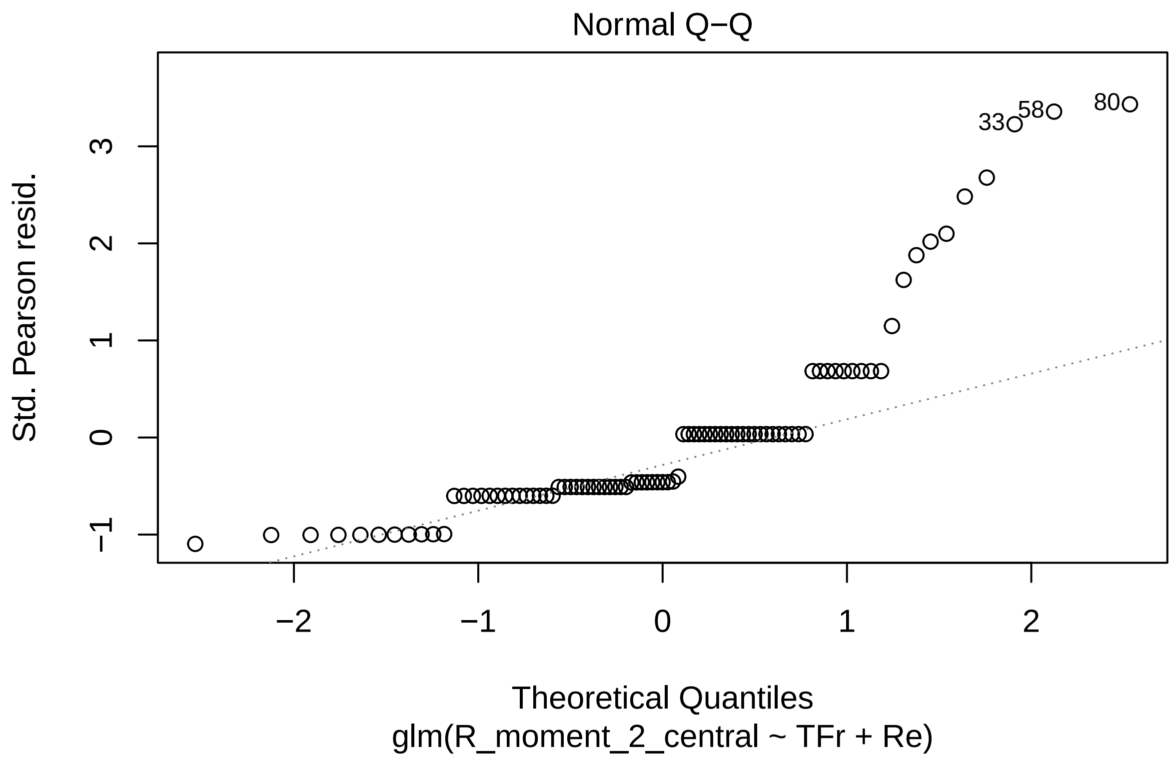
Linear fitting on central moments 2 through 4

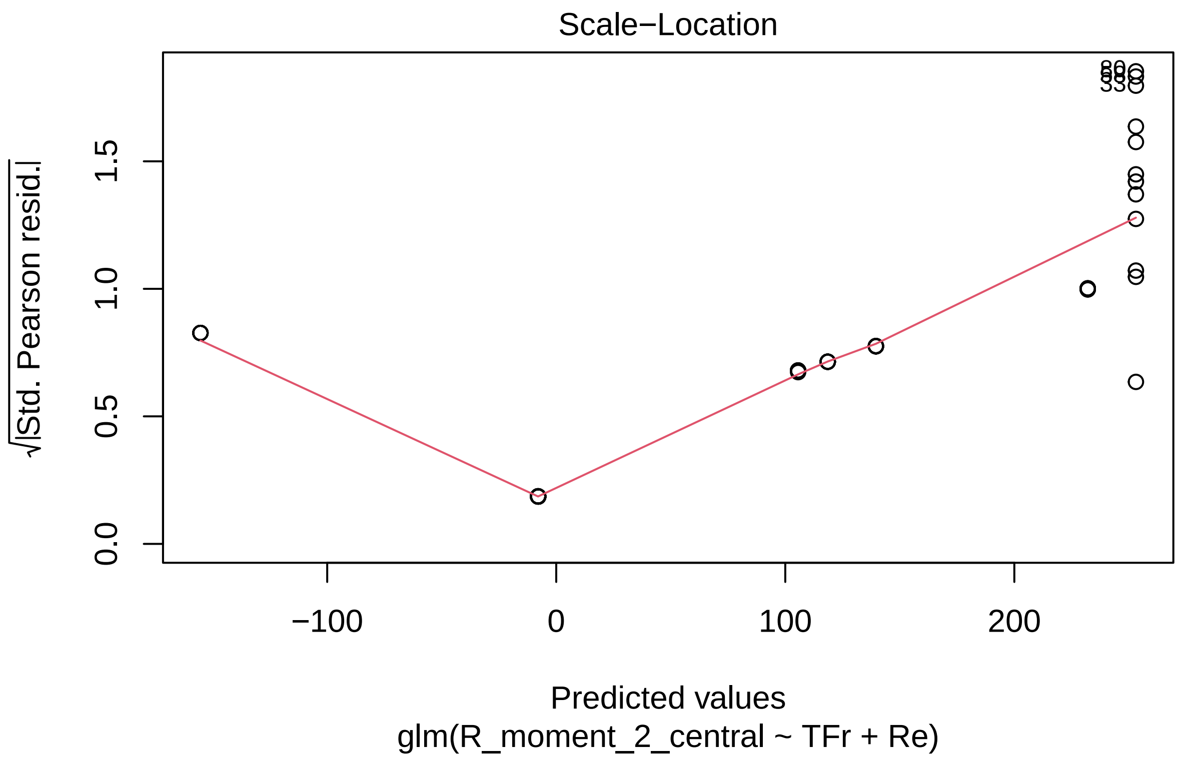
```
##
## Call:
## glm(formula = R_moment_2_central ~ TFr + Re, data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -252.57  -139.16  -104.99     7.98   791.18
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 299.6445    53.6449   5.586 2.68e-07 ***
## TFr         -10.2315     3.8471  -2.660 0.009332 **
## Re           -0.8472     0.2221  -3.815 0.000256 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 54789.33)
```

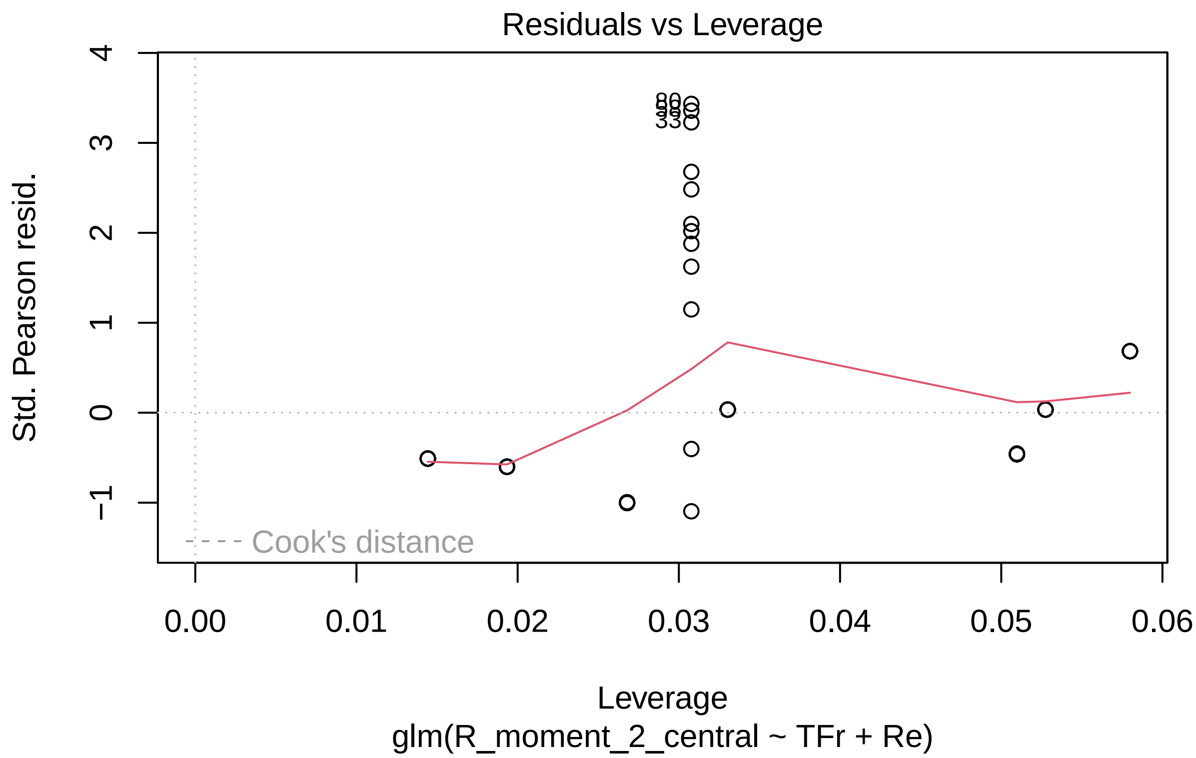
```
##
##     Null deviance: 6032130  on 88  degrees of freedom
## Residual deviance: 4711882  on 86  degrees of freedom
## AIC: 1228.6
##
## Number of Fisher Scoring iterations: 2
```

```
plot(step_full_linear_E2_central)
```




Normal Q−Q
Std. Pearson resid.
Theoretical Quantiles
glm(R_moment_2_central ~ TFr + Re)
33
58
80
−2
−1
0
1
2
3


Scale−Location
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
−100
0
100
200
Predicted values
glm(R_moment_2_central ~ TFr + Re)

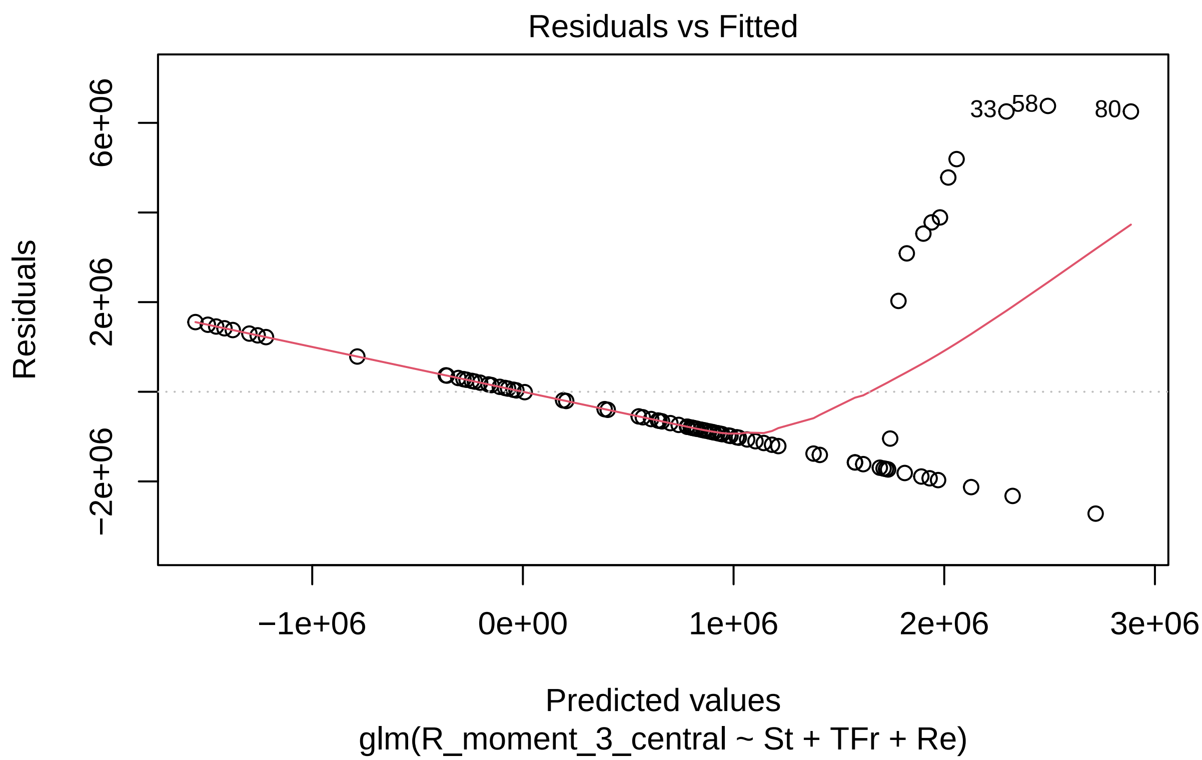


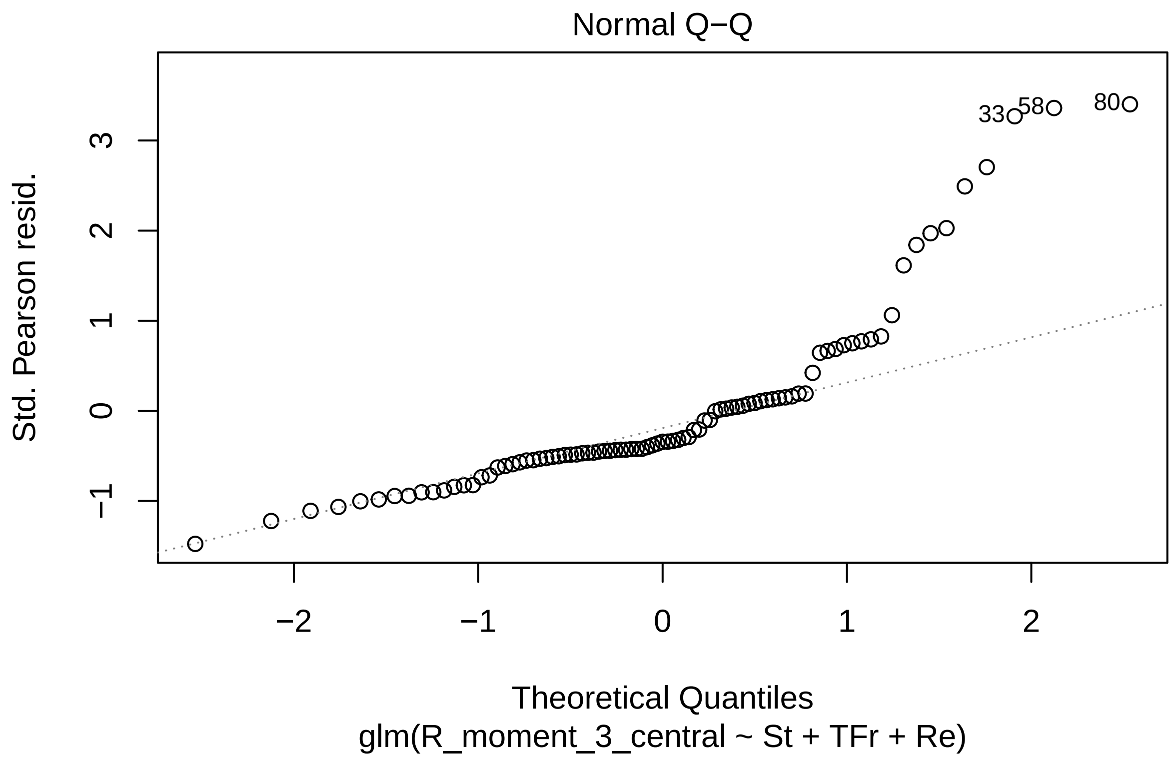

```r
full_linear_E3_central <- glm(R_moment_3_central ~ St + TFr + Re, data = data_train)
step_full_linear_E3_central <- stepAIC(full_linear_E3_central, direction = "both", trace = FALSE)
summary(step_full_linear_E3_central)
```

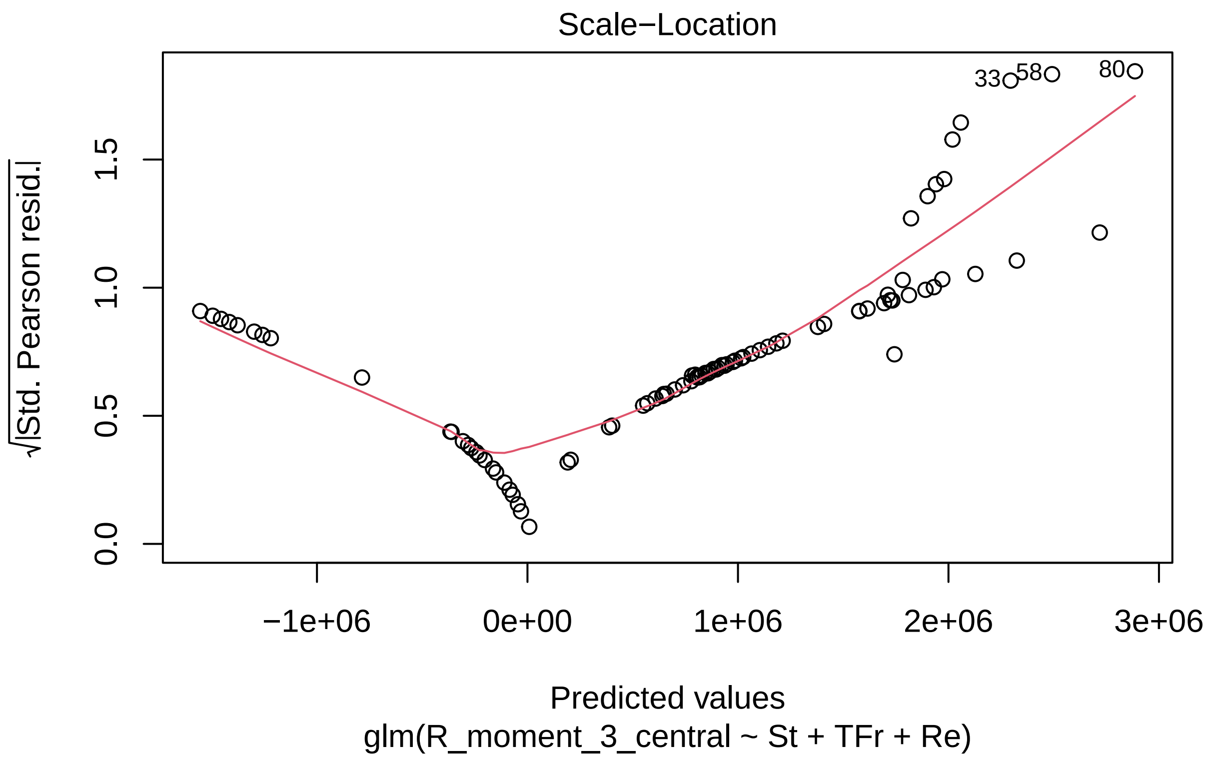
```
##
## Call:
## glm(formula = R_moment_3_central ~ St + TFr + Re, data = data_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2718640  -1024952   -660538    281872   6377459
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2082346     508033   4.099 9.46e-05 ***
## St            394050     264781   1.488 0.140396
## TFr           -81444      32075  -2.539 0.012933 *
## Re             -6831       1851  -3.691 0.000393 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 3.801057e+12)
##
##     Null deviance: 4.1934e+14  on 88  degrees of freedom
## Residual deviance: 3.2309e+14  on 85  degrees of freedom
```

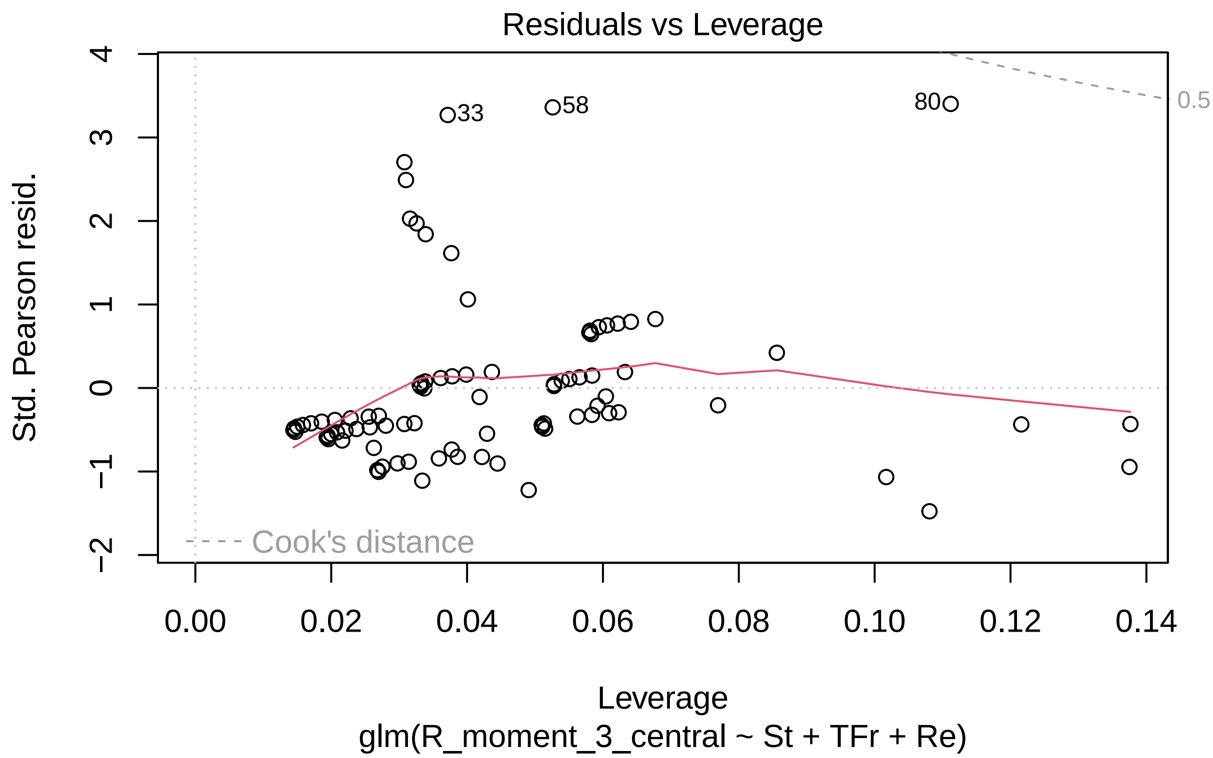
```
## AIC: 2836.5
##
## Number of Fisher Scoring iterations: 2
```

```
plot(step_full_linear_E3_central)
```




Normal Q−Q
Std. Pearson resid.
Theoretical Quantiles
glm(R_moment_3_central ~ St + TFr + Re)
33
58
80
−2
−1
0
1
2
3


Scale−Location
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
33
58
80
−1e+06
0e+00
1e+06
2e+06
3e+06
Predicted values
glm(R_moment_3_central ~ St + TFr + Re)

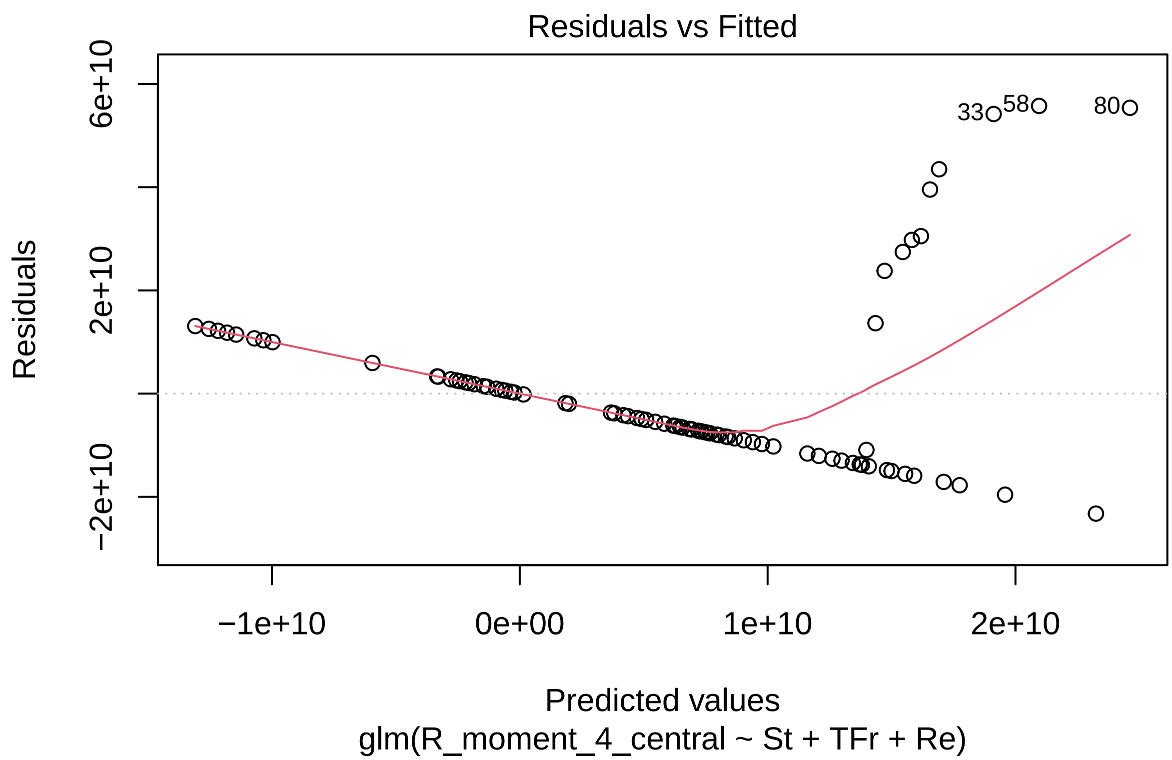


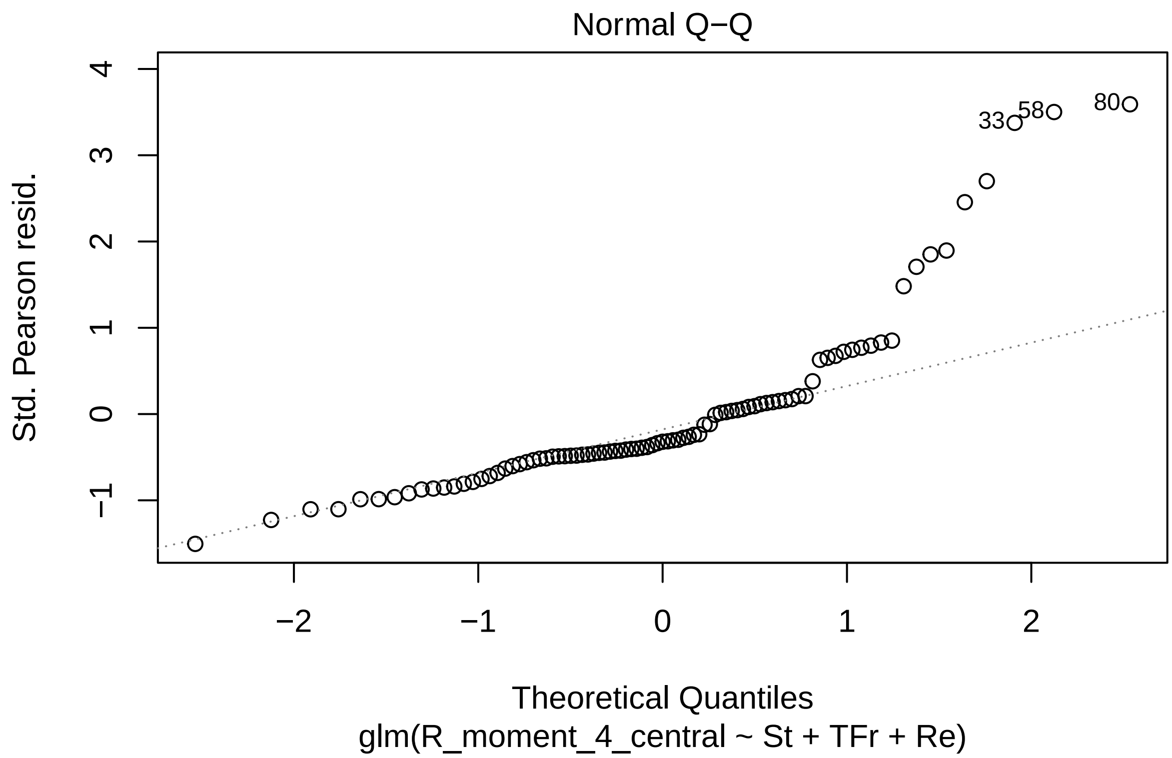

```r
full_linear_E4_central <- glm(R_moment_4_central ~ St + TFr + Re, data = data_train)
step_full_linear_E4_central <- stepAIC(full_linear_E4_central, direction = "both", trace = FALSE)
summary(step_full_linear_E4_central)
```

```
##
## Call:
## glm(formula = R_moment_4_central ~ St + TFr + Re, data = data_train)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.325e+10  -8.400e+09  -5.100e+09   2.554e+09   5.574e+10
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.673e+10  4.262e+09   3.925 0.000175 ***
## St           3.667e+09  2.222e+09   1.651 0.102522
## TFr         -6.673e+08  2.691e+08  -2.480 0.015126 *
## Re          -5.609e+07  1.553e+07  -3.612 0.000513 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2.675639e+20)
##
##     Null deviance: 2.9409e+22  on 88  degrees of freedom
## Residual deviance: 2.2743e+22  on 85  degrees of freedom
```

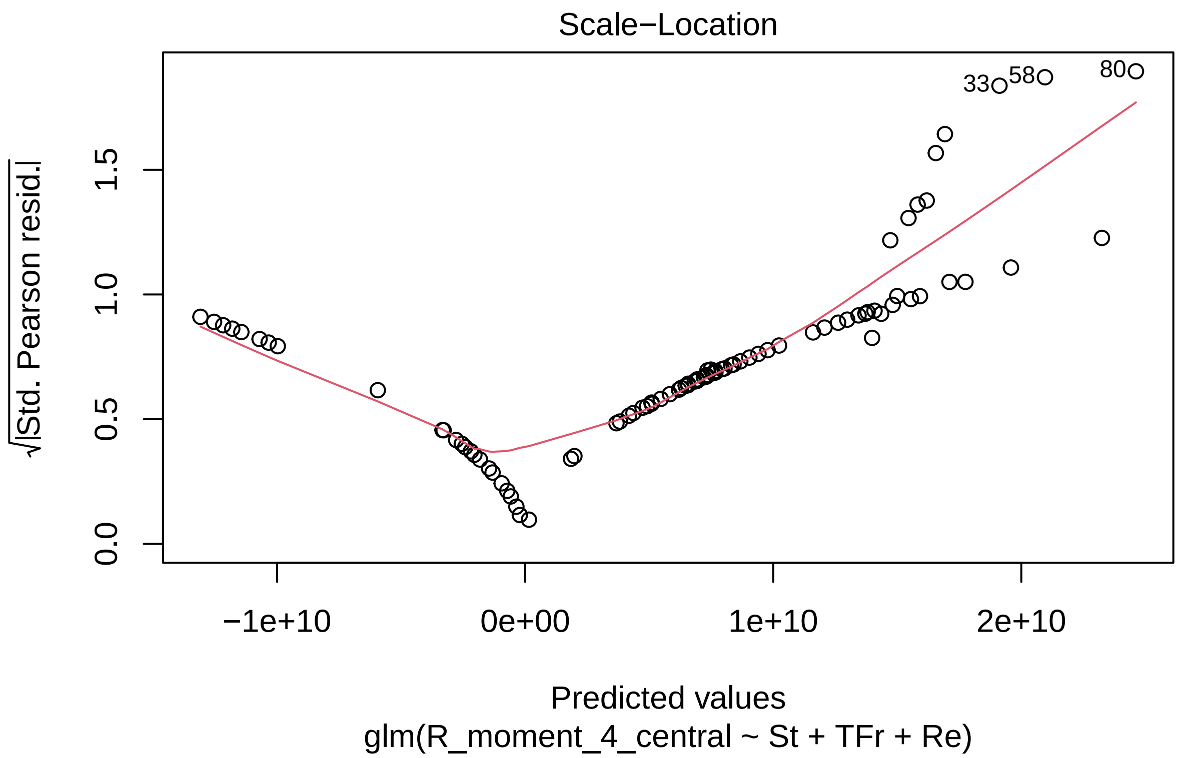
```
## AIC: 4444.7
##
## Number of Fisher Scoring iterations: 2
```

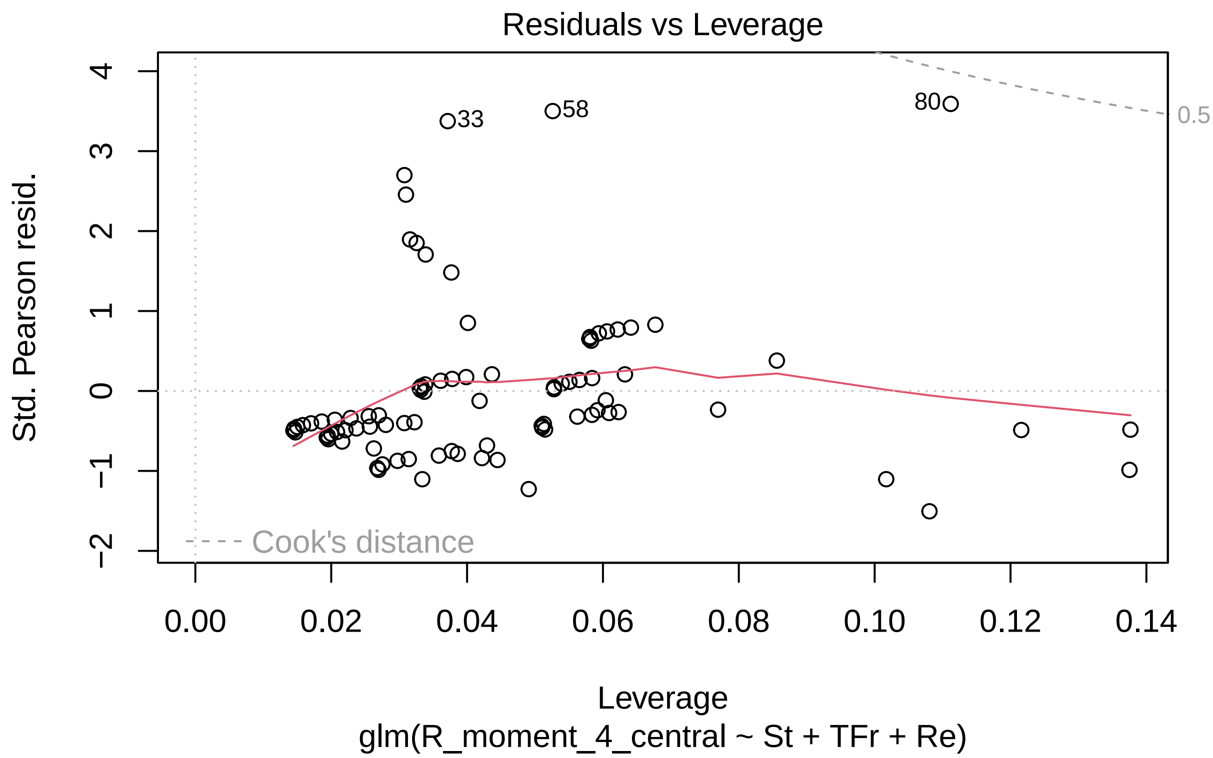
```
plot(step_full_linear_E4_central)
```





Normal Q−Q
Std. Pearson resid.
4
3
2
1
0
−1
−2
−1
0
1
2
33
58
80
Theoretical Quantiles
glm(R_moment_4_central ~ St + TFr + Re)


Scale−Location
33
58
80
1.5
1.0
0.5
0.0
√|Std. Pearson resid.|
−1e+10
0e+00
1e+10
2e+10
Predicted values
glm(R_moment_4_central ~ St + TFr + Re)



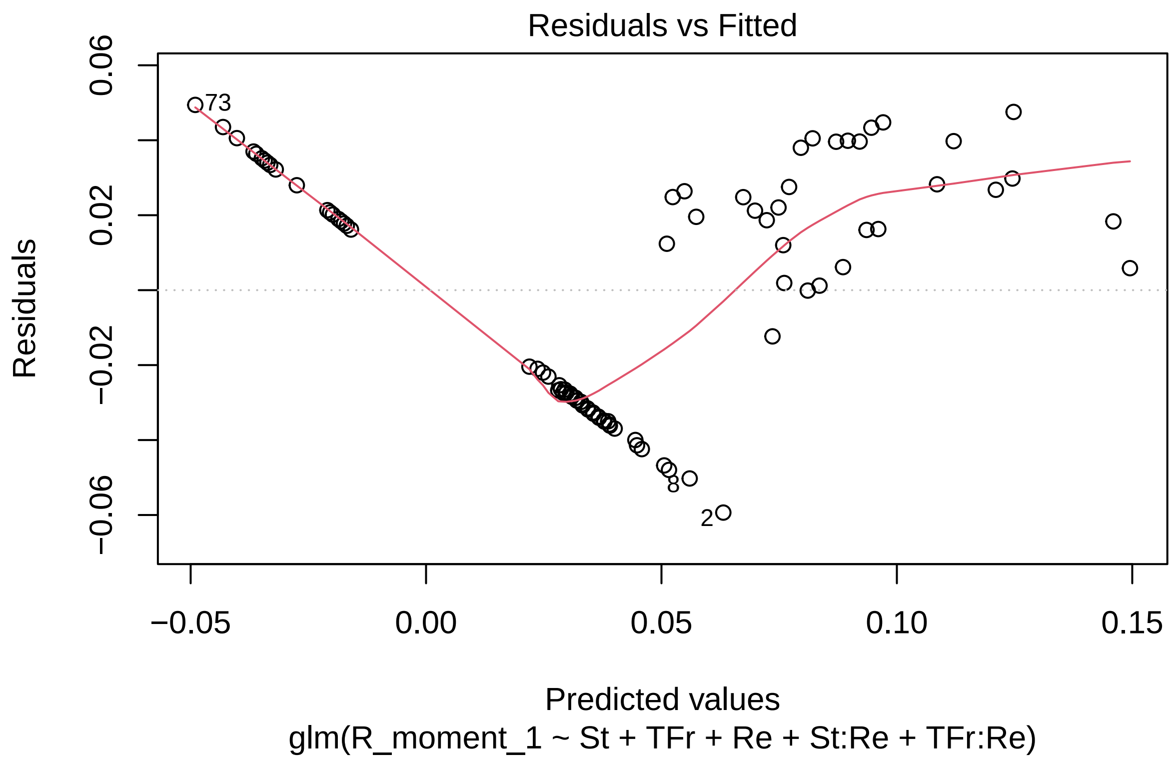

```
full_linear_interactions_E1 <- glm(R_moment_1 ~ St*TFr + St*Re + TFr*Re, data = data_train)
step_full_linear_interactions_E1 <- stepAIC(full_linear_interactions_E1, direction = "both", trace = FAl
summary(step_full_linear_interactions_E1)
```

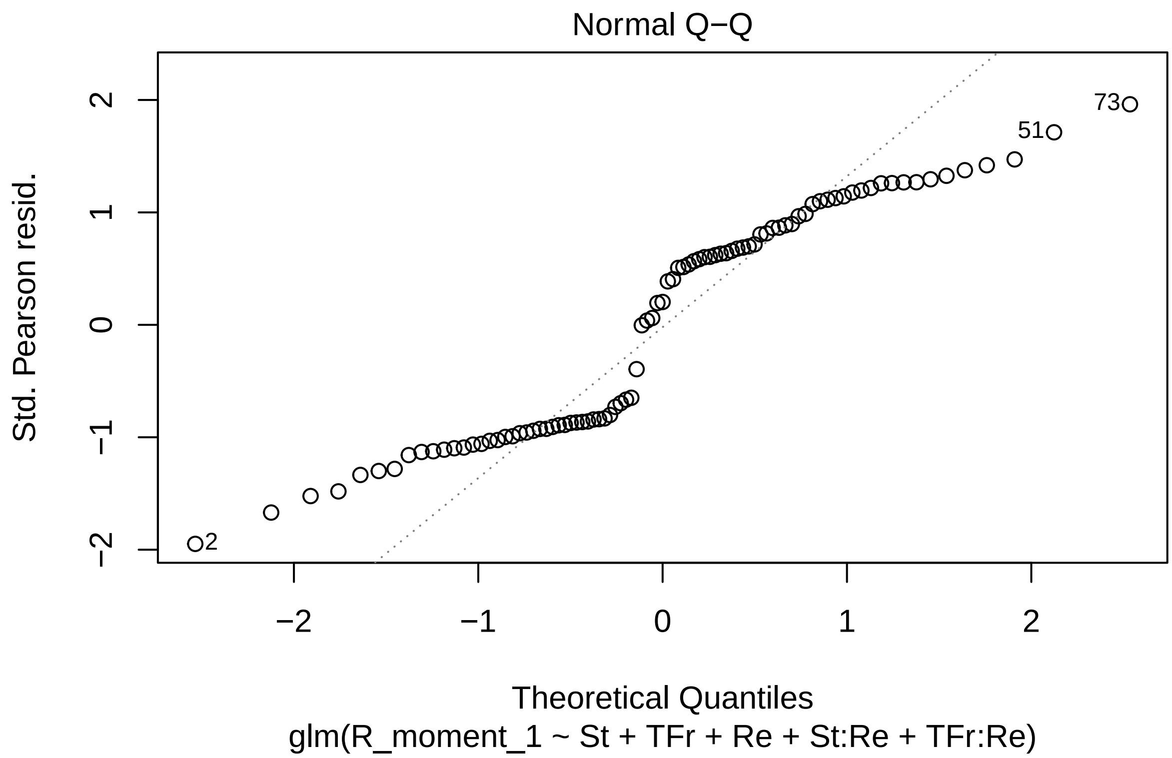
Best AiC model with interactions

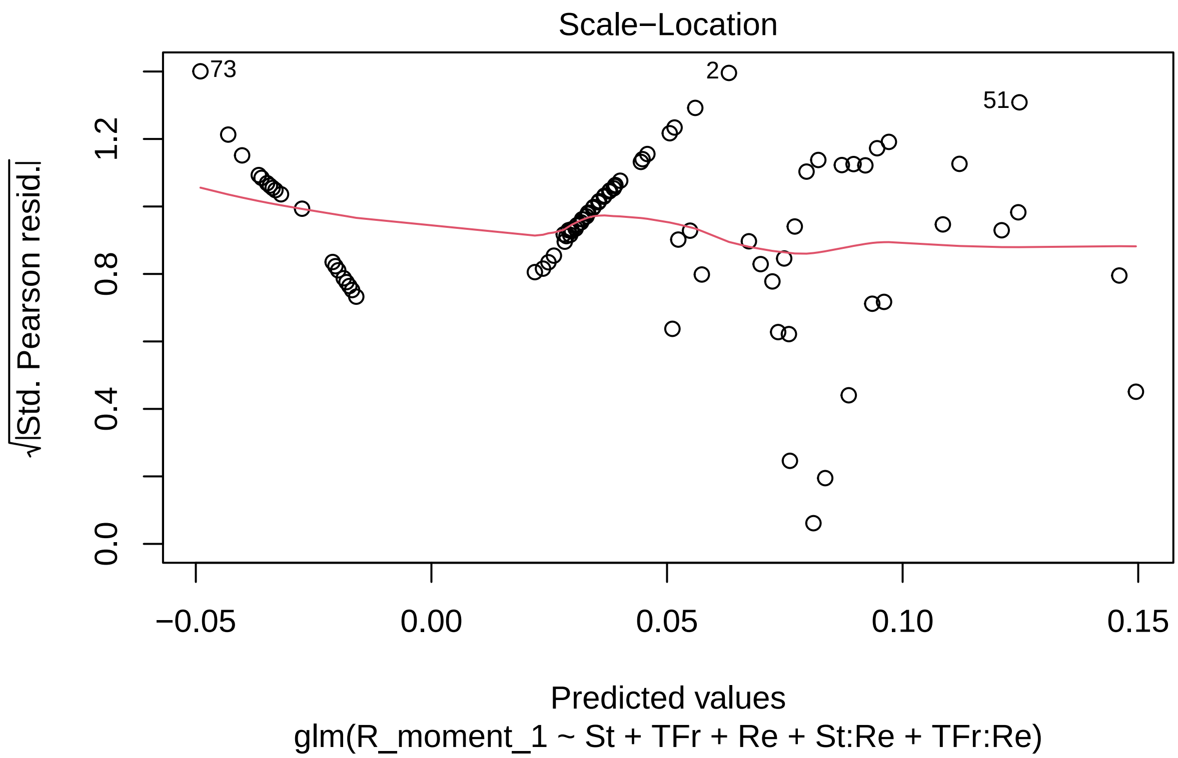
```
##
## Call:
## glm(formula = R_moment_1 ~ St + TFr + Re + St:Re + TFr:Re, data = data_train)
##
## Deviance Residuals:
##       Min         1Q     Median         3Q        Max
## -0.059348  -0.029496   0.006145   0.027529   0.049423
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.822e-02  1.140e-02   8.615 3.85e-13 ***
## St           3.398e-02  8.969e-03   3.789 0.000286 ***
## TFr         -2.534e-03  1.161e-03  -2.182 0.031927 *
## Re          -3.176e-04  4.925e-05  -6.448 7.08e-09 ***
## St:Re       -1.002e-04  3.899e-05  -2.570 0.011953 *
## TFr:Re       9.098e-06  4.559e-06   1.995 0.049275 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.001036755)
##
##     Null deviance: 0.274427  on 88  degrees of freedom
## Residual deviance: 0.086051  on 83  degrees of freedom
## AIC: -351.22
##
## Number of Fisher Scoring iterations: 2
```

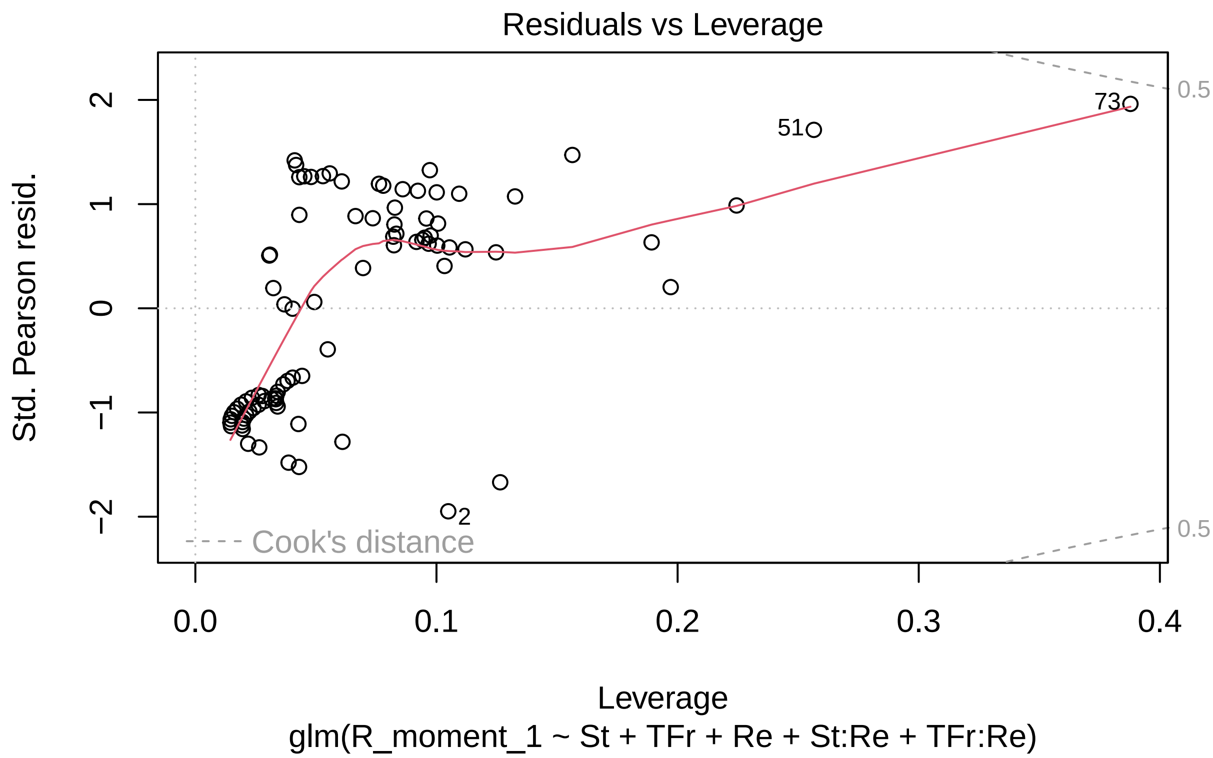
```
plot(step_full_linear_interactions_E1)
```




Normal Q−Q
Std. Pearson resid.
Theoretical Quantiles
glm(R_moment_1 ~ St + TFr + Re + St:Re + TFr:Re)
73
51
2
−2
−1
0
1
2



Scale−Location
73
2
51
1.2
0.8
0.4
0.0
√|Std. Pearson resid.|
−0.05
0.00
0.05
0.10
0.15
Predicted values
glm(R_moment_1 ~ St + TFr + Re + St:Re + TFr:Re)

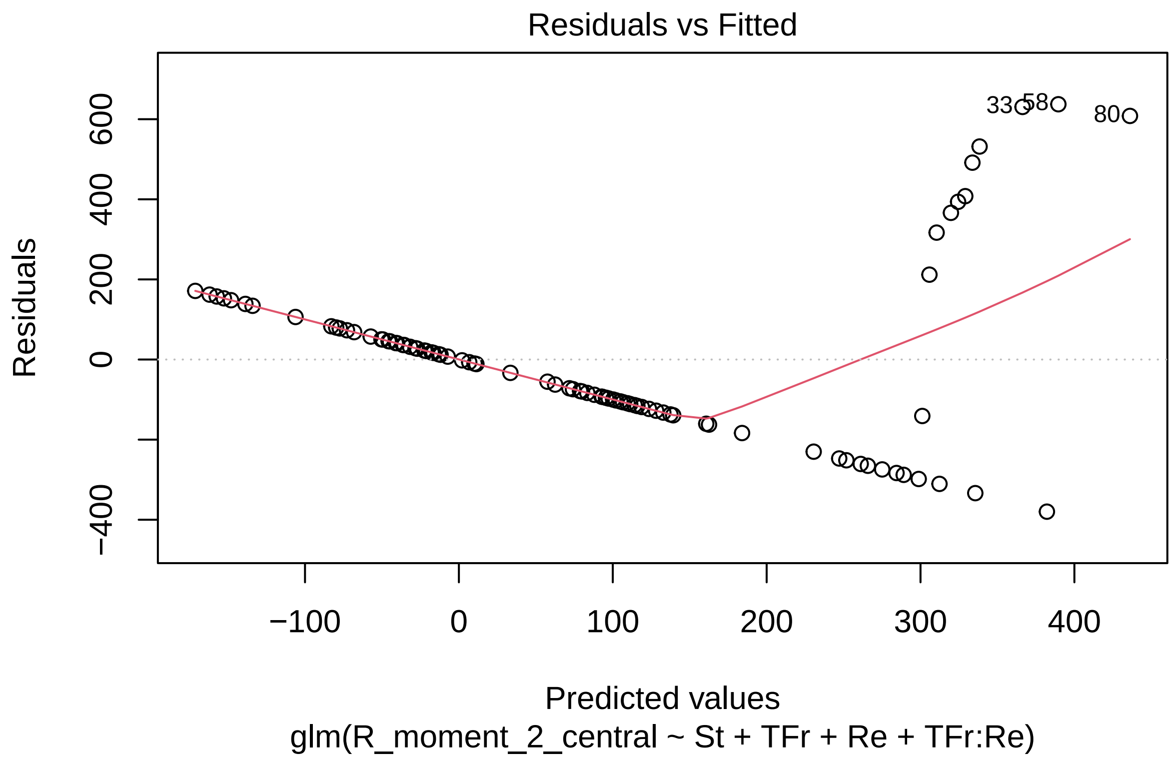


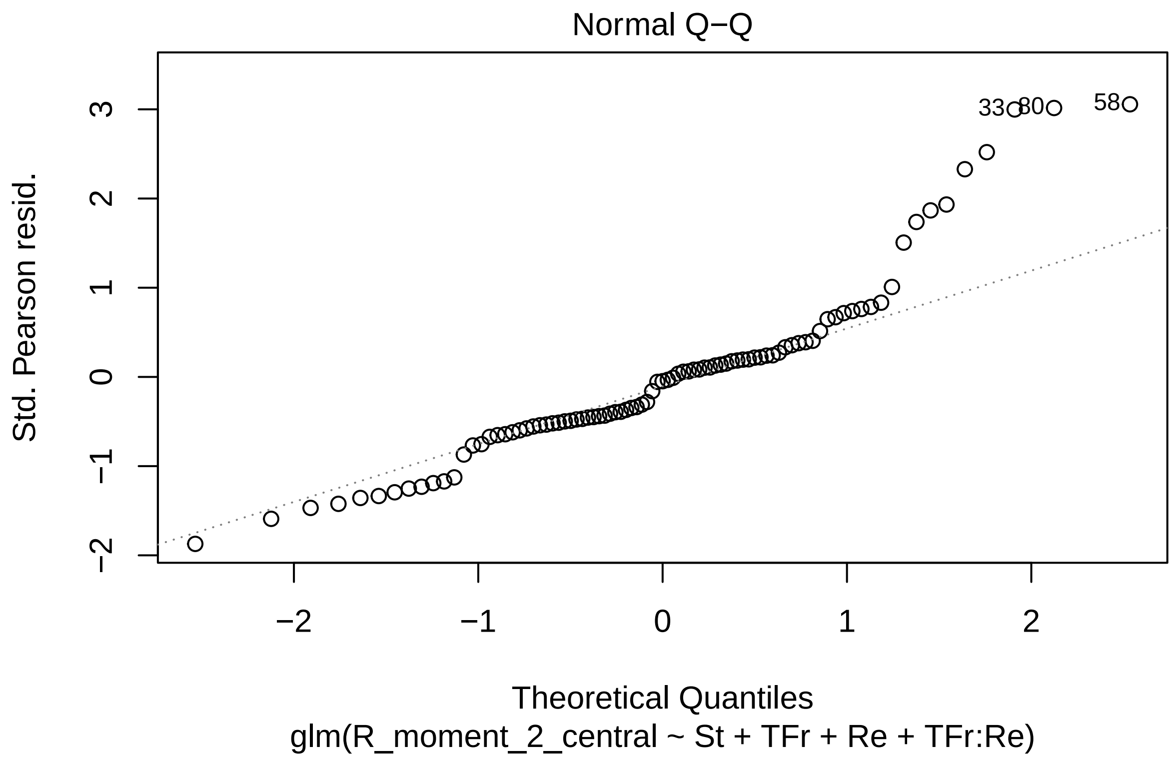

```
full_linear_interactions_E2 <- glm(R_moment_2_central ~ St*TFr + St*Re + TFr*Re, data = data_train)
step_full_linear_interactions_E2 <- stepAIC(full_linear_interactions_E2, direction = "both", trace = FAL
summary(step_full_linear_interactions_E2)
```

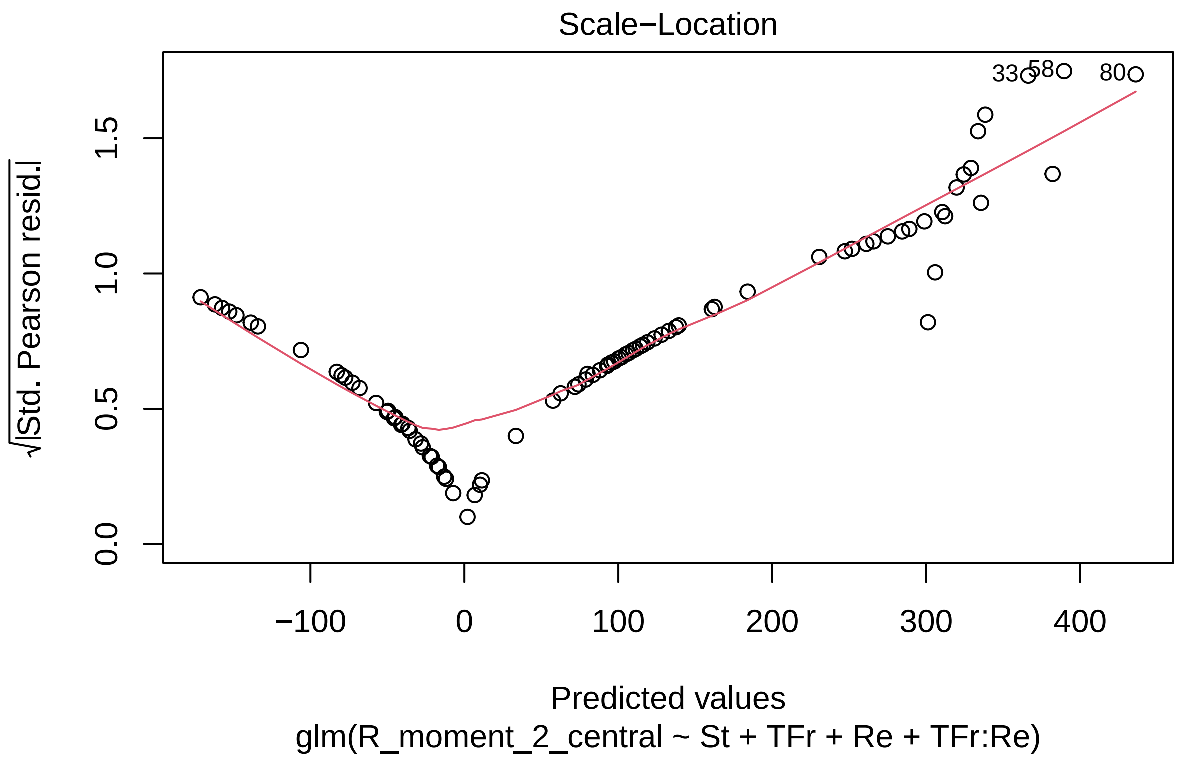
```
##
## Call:
## glm(formula = R_moment_2_central ~ St + TFr + Re + TFr:Re, data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -379.83  -115.96   -10.12    68.41   637.30
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 327.48973   58.60963   5.588 2.78e-07 ***
## St           46.54204   29.30260   1.588 0.115970
## TFr         -36.88006    7.71023  -4.783 7.29e-06 ***
## Re           -1.19141    0.22338  -5.333 8.00e-07 ***
## TFr:Re        0.11802    0.03007   3.925 0.000177 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 46461.98)
##
##     Null deviance: 6032130  on 88  degrees of freedom
```

```
## Residual deviance: 3902807  on 84  degrees of freedom
## AIC: 1215.9
##
## Number of Fisher Scoring iterations: 2
```

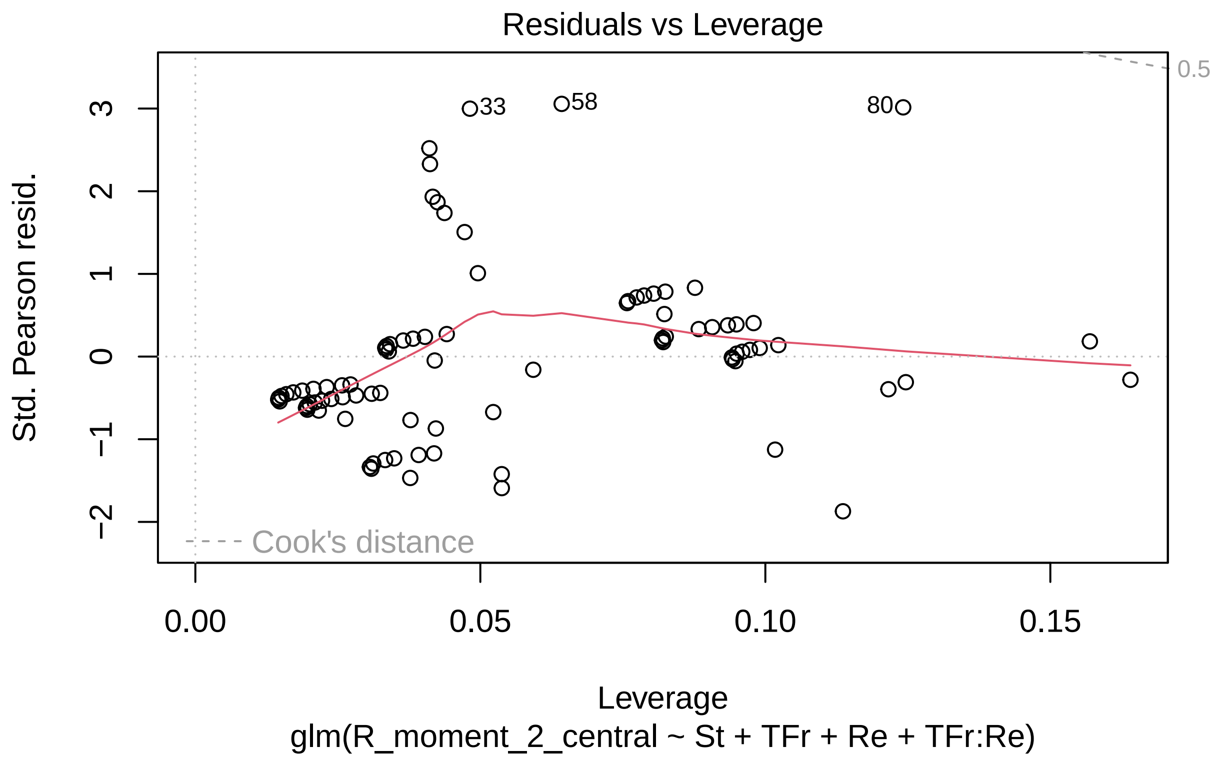
```
plot(step_full_linear_interactions_E2)
```




Normal Q−Q
Std. Pearson resid.
3
2
1
0
−1
−2
33
80
58
−2
−1
0
1
2
Theoretical Quantiles
glm(R_moment_2_central ~ St + TFr + Re + TFr:Re)



Scale−Location
33
58
80
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
−100
0
100
200
300
400
Predicted values
glm(R_moment_2_central ~ St + TFr + Re + TFr:Re)

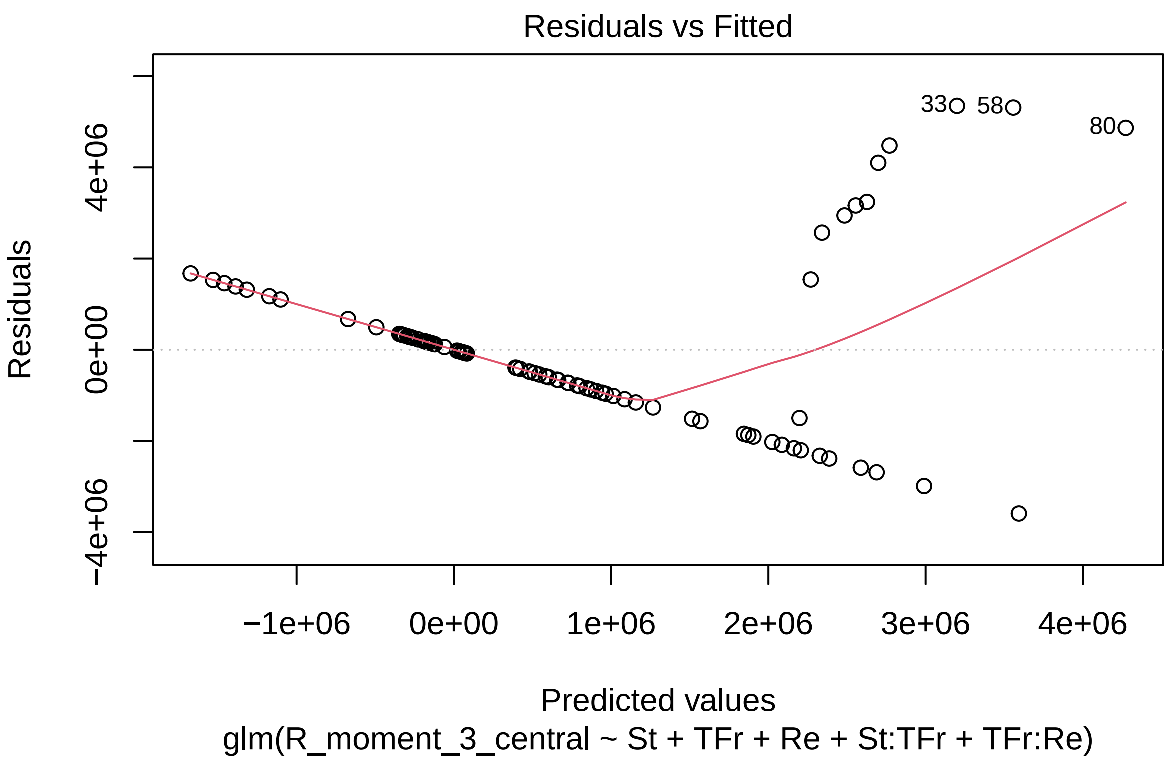


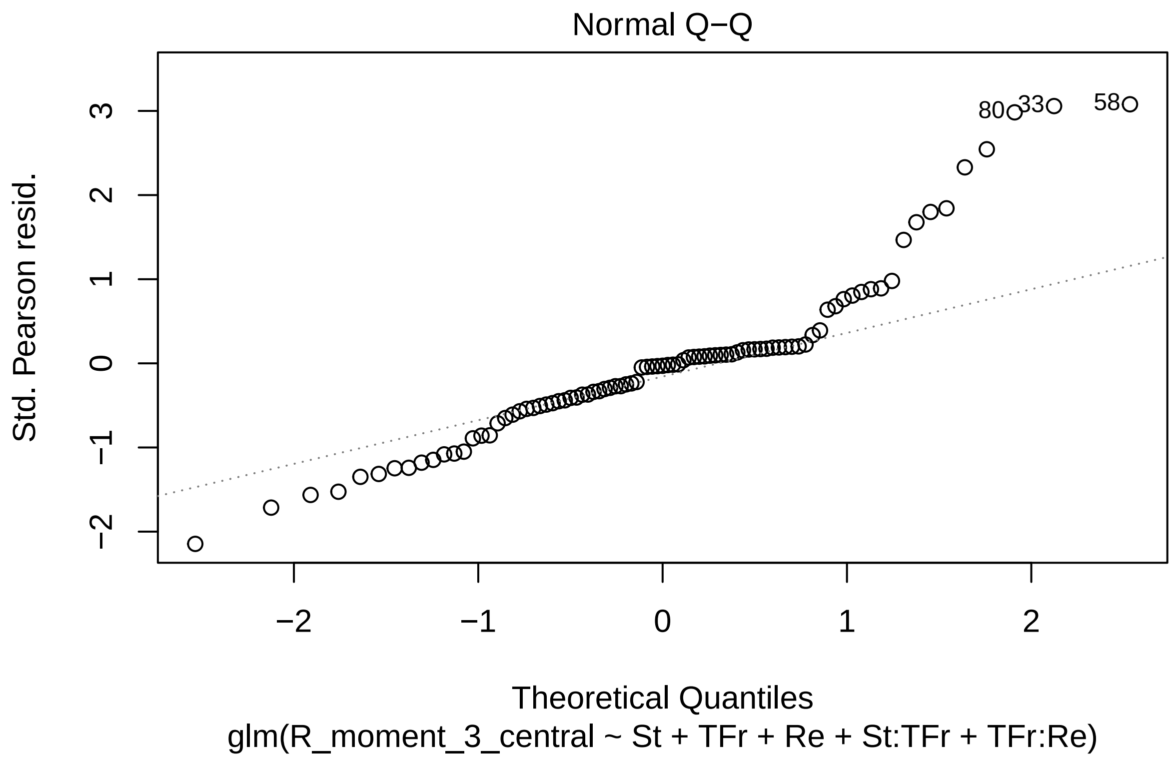

```
full_linear_interactions_E3 <- glm(R_moment_3_central ~ St*TFr + St*Re + TFr*Re, data = data_train)
step_full_linear_interactions_E3 <- stepAIC(full_linear_interactions_E3, direction = "both", trace = FAl
summary(step_full_linear_interactions_E3)
```

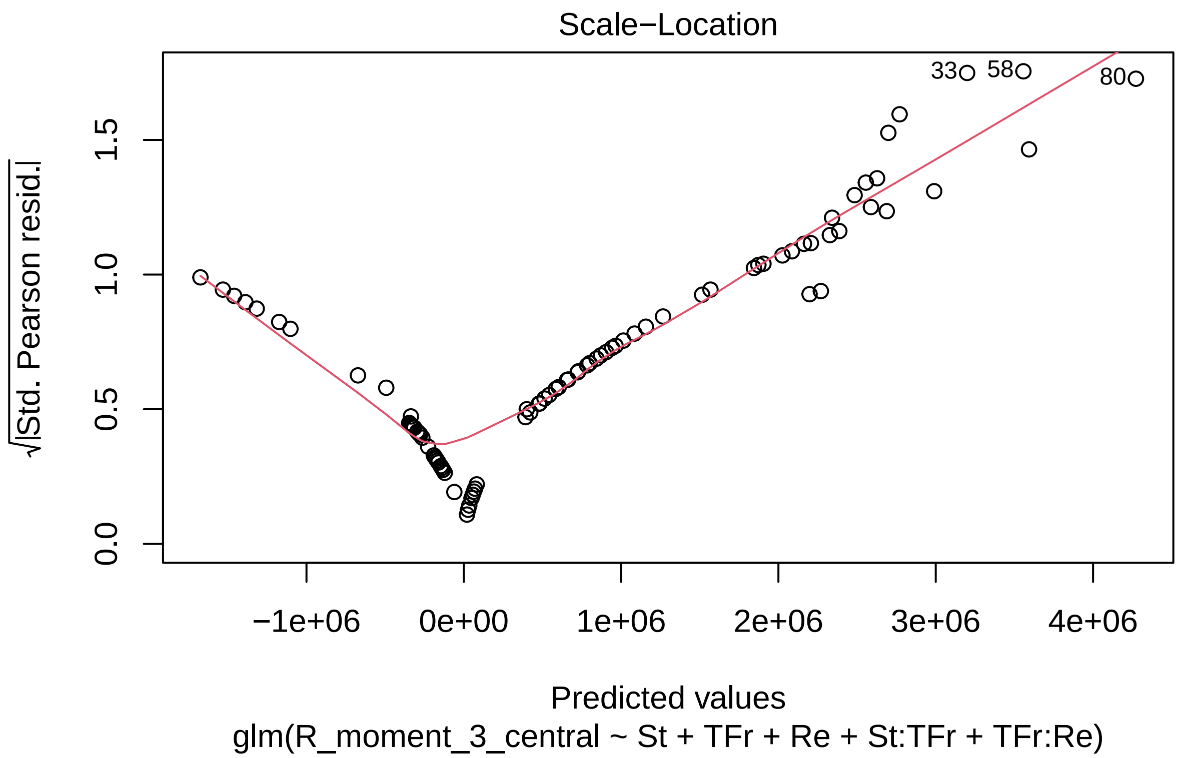
```
##
## Call:
## glm(formula = R_moment_3_central ~ St + TFr + Re + St:TFr + TFr:Re,
##     data = data_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3592944   -904717    -49411    332389   5349989
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2525572.0   493187.5   5.121 1.94e-06 ***
## St           556593.3   258219.6   2.156 0.034018 *
## TFr         -252297.4    72716.5  -3.470 0.000829 ***
## Re            -9805.2     1864.1  -5.260 1.10e-06 ***
## St:TFr       -54689.6    37680.2  -1.451 0.150434
## TFr:Re          953.2      250.9   3.798 0.000276 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 3.231922e+12)
```

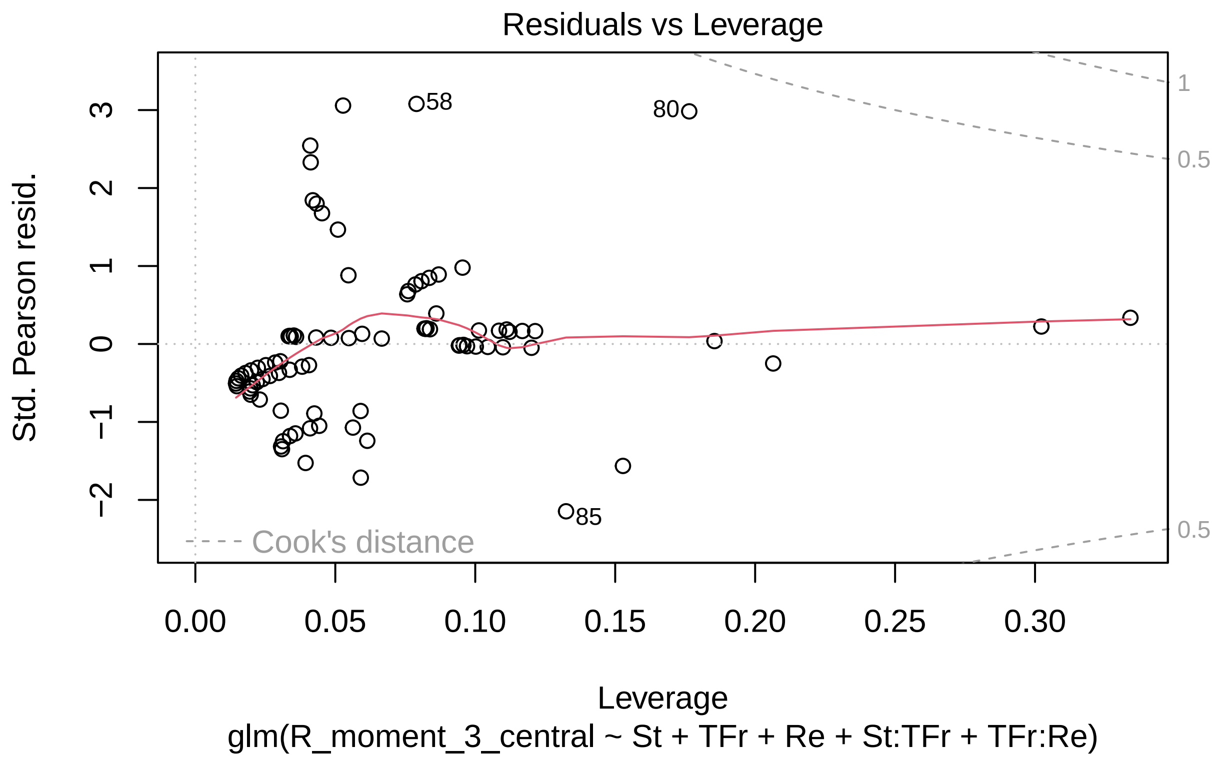
```
##
##     Null deviance: 4.1934e+14  on 88  degrees of freedom
## Residual deviance: 2.6825e+14  on 83  degrees of freedom
## AIC: 2823.9
##
## Number of Fisher Scoring iterations: 2
```

```
plot(step_full_linear_interactions_E3)
```




Normal Q−Q
Std. Pearson resid.
Theoretical Quantiles
glm(R_moment_3_central ~ St + TFr + Re + St:TFr + TFr:Re)
80
33
58
−2
−1
0
1
2
3


Scale−Location
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
33
58
80
−1e+06
0e+00
1e+06
2e+06
3e+06
4e+06
Predicted values
glm(R_moment_3_central ~ St + TFr + Re + St:TFr + TFr:Re)

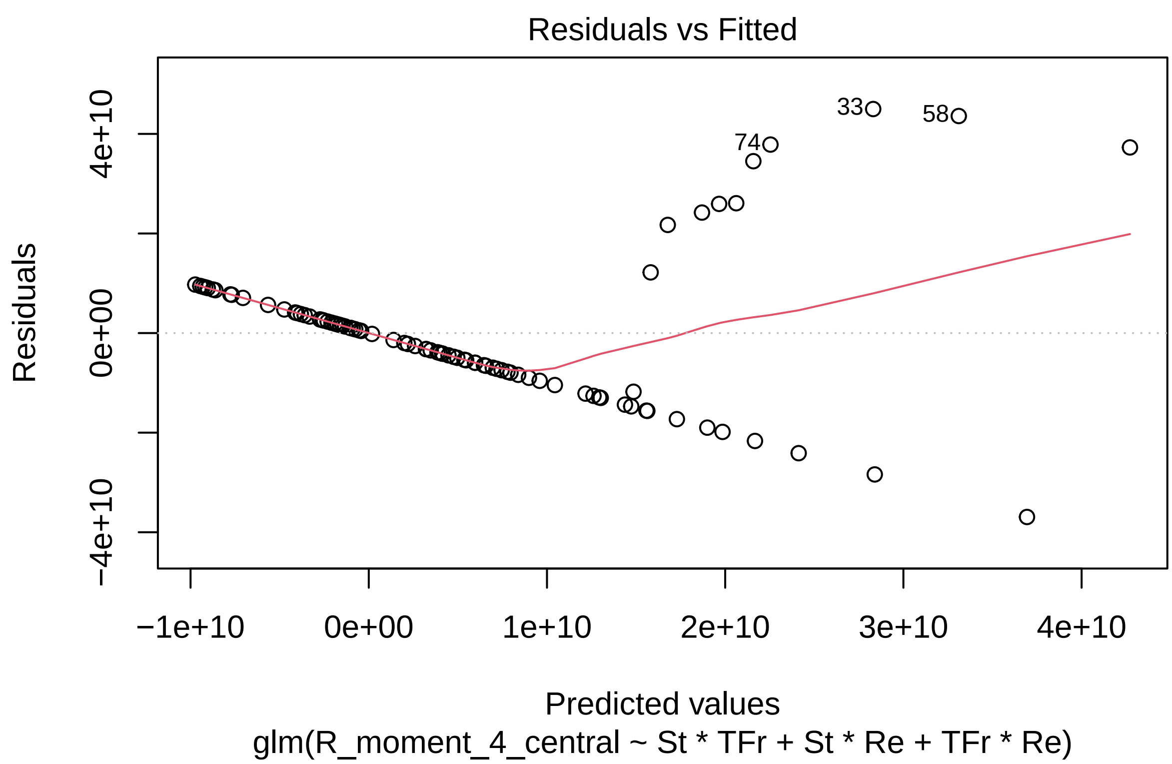


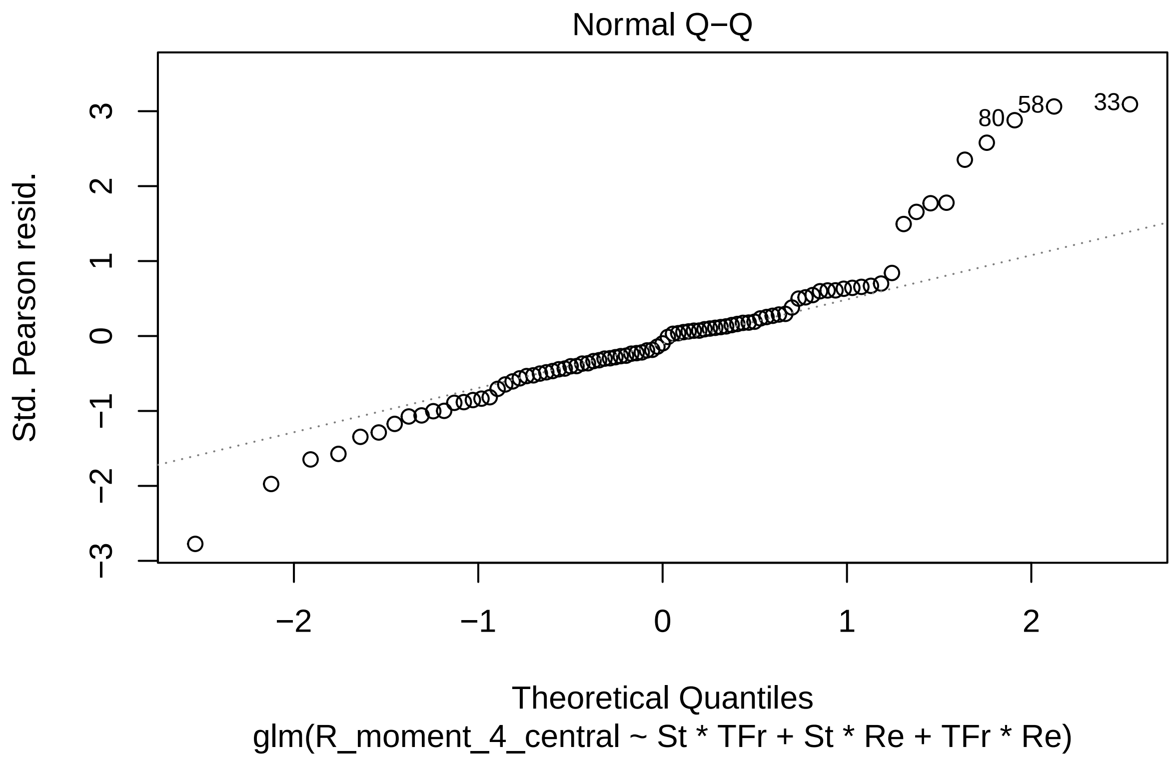

```r
full_linear_interactions_E4 <- glm(R_moment_4_central ~ St*TFr + St*Re + TFr*Re, data = data_train)
step_full_linear_interactions_E4 <- stepAIC(full_linear_interactions_E4, direction = "both", trace = FA
summary(step_full_linear_interactions_E4)
```

```
##
## Call:
## glm(formula = R_moment_4_central ~ St * TFr + St * Re + TFr *
##     Re, data = data_train)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -3.694e+10  -7.457e+09  -1.392e+09   4.131e+09   4.499e+10
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.528e+10  5.349e+09   2.857 0.005418 **
## St           1.050e+10  4.256e+09   2.467 0.015707 *
## TFr         -1.915e+09  6.113e+08  -3.133 0.002401 **
## Re          -5.598e+07  2.293e+07  -2.442 0.016773 *
## St:TFr      -5.176e+08  3.144e+08  -1.646 0.103499
## St:Re       -2.662e+07  1.816e+07  -1.466 0.146598
## TFr:Re       7.303e+06  2.125e+06   3.438 0.000924 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

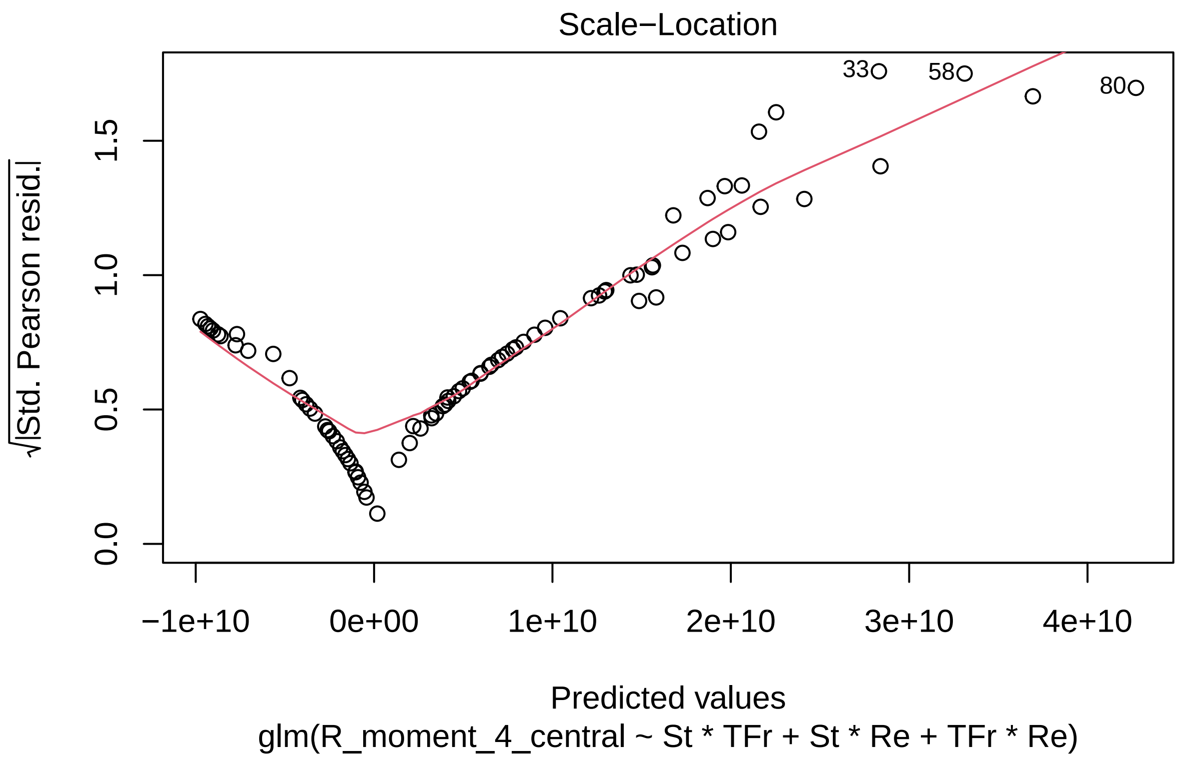
```
## (Dispersion parameter for gaussian family taken to be 2.246926e+20)
##
##     Null deviance: 2.9409e+22  on 88  degrees of freedom
## Residual deviance: 1.8425e+22  on 82  degrees of freedom
## AIC: 4431.9
##
## Number of Fisher Scoring iterations: 2
```

```
plot(step_full_linear_interactions_E4)
```

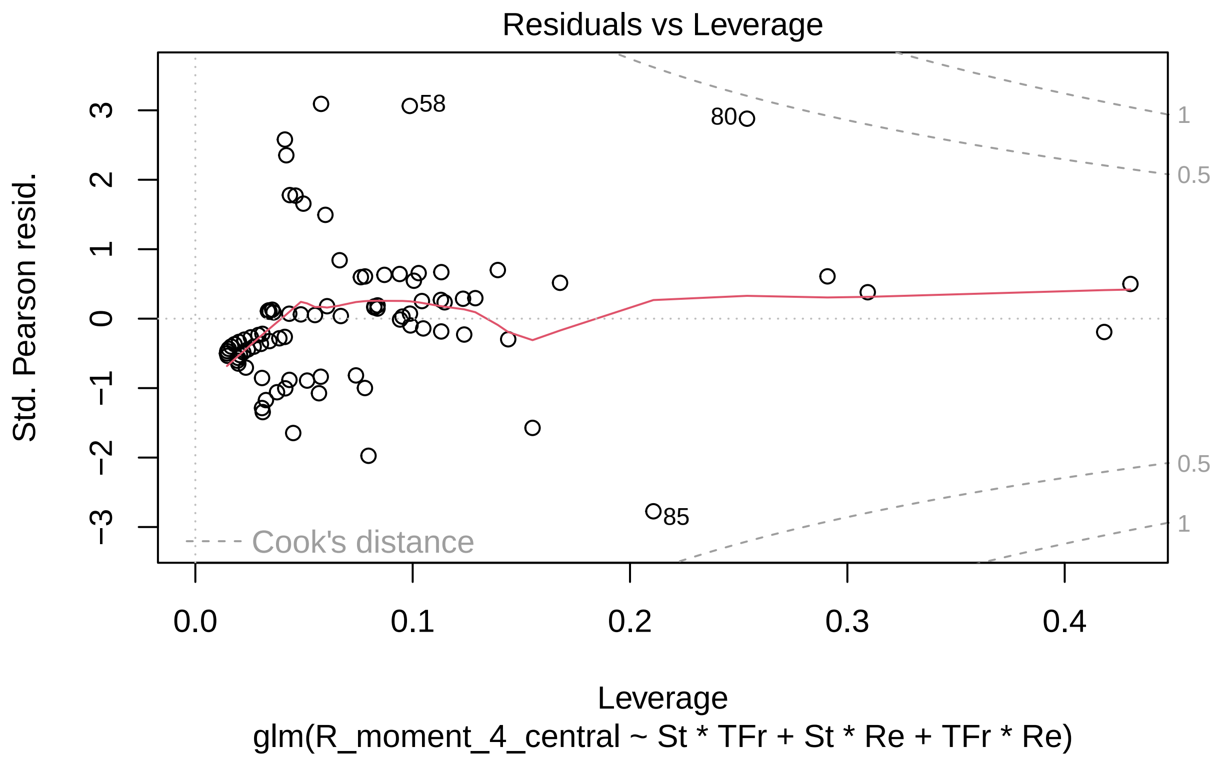




Normal Q−Q
Std. Pearson resid.
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
80
58
33
Theoretical Quantiles
glm(R_moment_4_central ~ St * TFr + St * Re + TFr * Re)



Scale−Location
33
58
80
0.0
0.5
1.0
1.5
√|Std. Pearson resid.|
−1e+10
0e+00
1e+10
2e+10
3e+10
4e+10
Predicted values
glm(R_moment_4_central ~ St * TFr + St * Re + TFr * Re)




```r
library(boot)
cve_linear_E1 <- cv.glm(data_train, step_full_linear_E1, K=10)
cve_linear_E1$delta
```

Model Evaluation (Linear)

```
## [1] 0.001251509 0.001244488
```

```r
cve_linear_interactions_E1 <- cv.glm(data_train, step_full_linear_interactions_E1, K = 10)
cve_linear_interactions_E1$delta
```

```
## [1] 0.001257243 0.001239236
```

```r
cve_linear_E2 <- cv.glm(data_train, step_full_linear_E2_central, K=10)
cve_linear_E2$delta
```

```
## [1] 59566.67 59209.99
```

```r
cve_linear_interactions_E2 <- cv.glm(data_train, step_full_linear_E2_central, K = 10)
cve_linear_interactions_E2$delta
```

```
## [1] 55388.66 55258.21
```

```r
cve_linear_E3 <- cv.glm(data_train, step_full_linear_E3_central, K=10)
cve_linear_E3$delta
```

```
## [1] 4.161659e+12 4.133140e+12
```

```r
cve_linear_interactions_E3 <- cv.glm(data_train, step_full_linear_interactions_E3, K = 10)
cve_linear_interactions_E3$delta
```

```
## [1] 3.442602e+12 3.418975e+12
```

```r
cve_linear_E4 <- cv.glm(data_train, step_full_linear_E4_central, K=10)
cve_linear_E4$delta
```

```
## [1] 2.875863e+20 2.858526e+20
```

```r
cve_linear_interactions_E4 <- cv.glm(data_train, step_full_linear_interactions_E4, K = 10)
cve_linear_interactions_E4$delta
```

```
## [1] 2.314118e+20 2.300873e+20
```

It seems that the interactions are increasingly important. They are less important for the first and second moments. In fact, cross validation error increases for the second moment when interactions are added into the model. However, for the third through fourth moments, there is a pretty significant decrease in cross validation error when comparing the strictly linear models versus the ones with interactions.

This I think the linear models worth sharing with our physicist colleagues are the following:

```r
summary(step_full_linear_E1)
```

```
##
## Call:
## glm(formula = R_moment_1 ~ St + Re, data = data_train)
##
## Deviance Residuals:
##       Min         1Q     Median         3Q        Max
## -0.061936  -0.030347  -0.000174   0.034491   0.055714
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.102e-01  8.837e-03  12.475  < 2e-16 ***
## St           1.353e-02  4.621e-03   2.927  0.00438 **
## Re          -3.798e-04  3.215e-05 -11.816  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.001160225)
##
##     Null deviance: 0.274427  on 88  degrees of freedom
## Residual deviance: 0.099779  on 86  degrees of freedom
## AIC: -344.04
##
## Number of Fisher Scoring iterations: 2
```

```r
summary(step_full_linear_E2_central)
```

```
##
## Call:
## glm(formula = R_moment_2_central ~ TFr + Re, data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -252.57  -139.16  -104.99     7.98   791.18
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 299.6445    53.6449   5.586 2.68e-07 ***
## TFr         -10.2315     3.8471  -2.660 0.009332 **
## Re           -0.8472     0.2221  -3.815 0.000256 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 54789.33)
##
##     Null deviance: 6032130  on 88  degrees of freedom
## Residual deviance: 4711882  on 86  degrees of freedom
## AIC: 1228.6
##
## Number of Fisher Scoring iterations: 2
```

```r
summary(step_full_linear_interactions_E3)
```

```
##
## Call:
## glm(formula = R_moment_3_central ~ St + TFr + Re + St:TFr + TFr:Re,
##     data = data_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3592944   -904717    -49411    332389   5349989
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2525572.0   493187.5   5.121 1.94e-06 ***
## St           556593.3   258219.6   2.156 0.034018 *
## TFr         -252297.4    72716.5  -3.470 0.000829 ***
## Re            -9805.2     1864.1  -5.260 1.10e-06 ***
## St:TFr       -54689.6    37680.2  -1.451 0.150434
## TFr:Re          953.2      250.9   3.798 0.000276 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 3.231922e+12)
##
##     Null deviance: 4.1934e+14  on 88  degrees of freedom
## Residual deviance: 2.6825e+14  on 83  degrees of freedom
```

```
## AIC: 2823.9
##
## Number of Fisher Scoring iterations: 2
```

```
summary(step_full_linear_interactions_E4)
```

```
##
## Call:
## glm(formula = R_moment_4_central ~ St * TFr + St * Re + TFr *
##     Re, data = data_train)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -3.694e+10  -7.457e+09  -1.392e+09   4.131e+09   4.499e+10
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.528e+10  5.349e+09   2.857 0.005418 **
## St           1.050e+10  4.256e+09   2.467 0.015707 *
## TFr         -1.915e+09  6.113e+08  -3.133 0.002401 **
## Re          -5.598e+07  2.293e+07  -2.442 0.016773 *
## St:TFr      -5.176e+08  3.144e+08  -1.646 0.103499
## St:Re       -2.662e+07  1.816e+07  -1.466 0.146598
## TFr:Re       7.303e+06  2.125e+06   3.438 0.000924 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2.246926e+20)
##
##     Null deviance: 2.9409e+22  on 88  degrees of freedom
## Residual deviance: 1.8425e+22  on 82  degrees of freedom
## AIC: 4431.9
##
## Number of Fisher Scoring iterations: 2
```

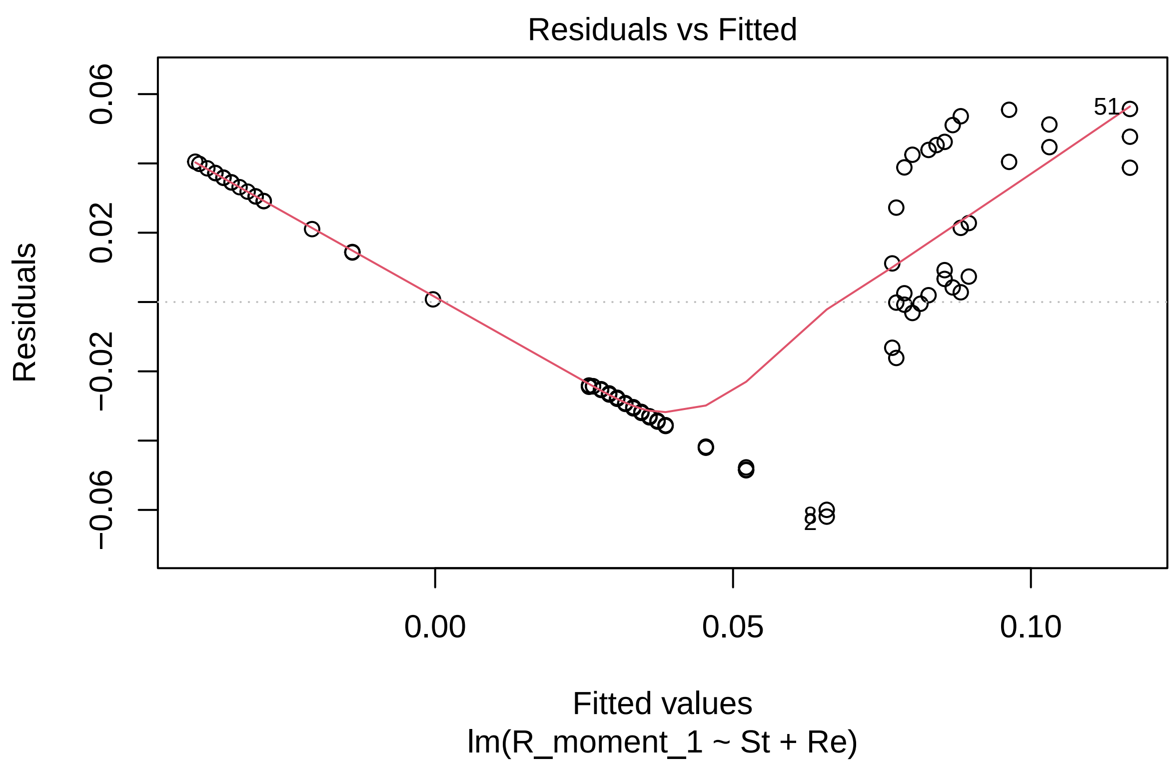
OR (by calling lm version of the functions)

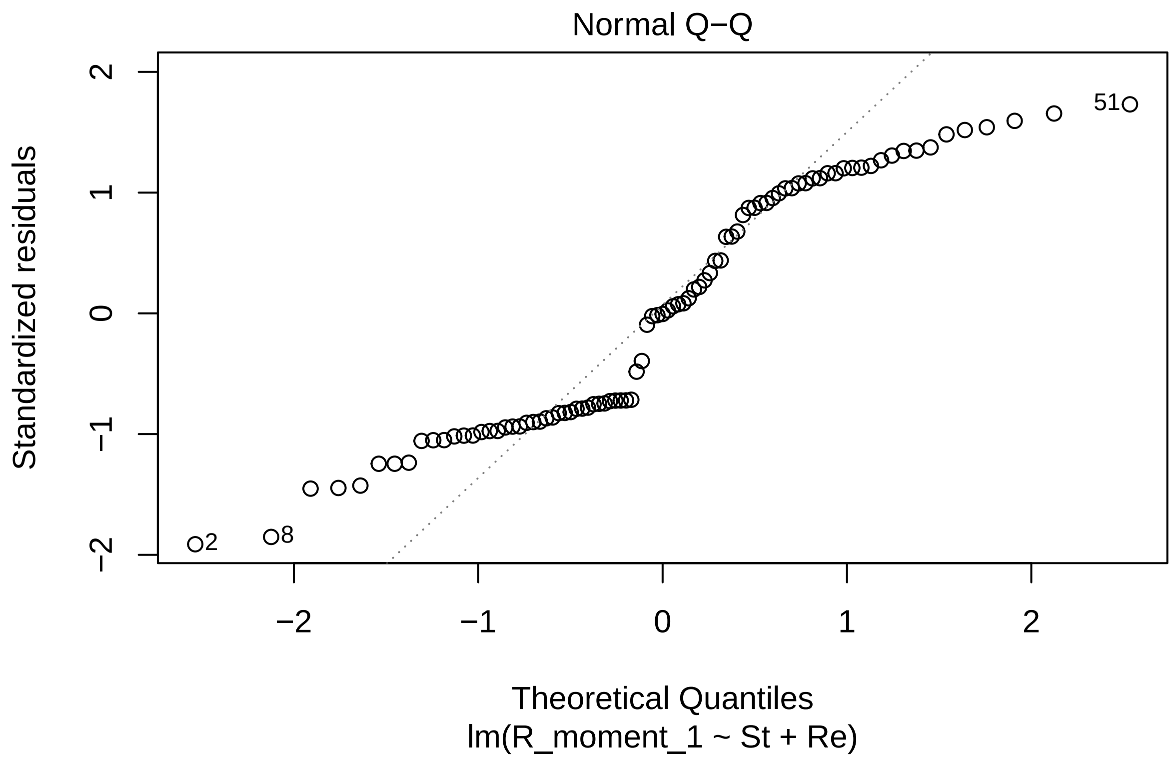
```
lm_fit_E1 <- lm(R_moment_1 ~ St + Re, data = data_train)
summary(lm_fit_E1)
```

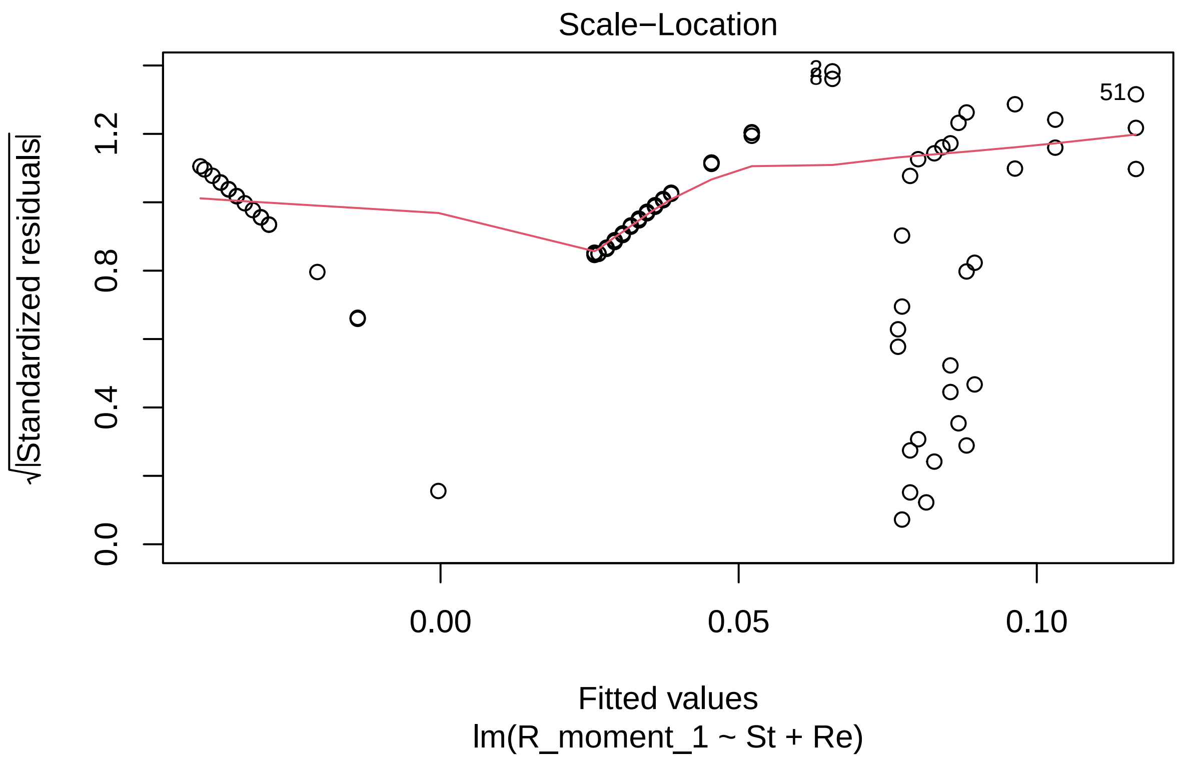
```
##
## Call:
## lm(formula = R_moment_1 ~ St + Re, data = data_train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.061936 -0.030347 -0.000174  0.034491  0.055714
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.102e-01  8.837e-03  12.475  < 2e-16 ***
## St           1.353e-02  4.621e-03   2.927  0.00438 **
```

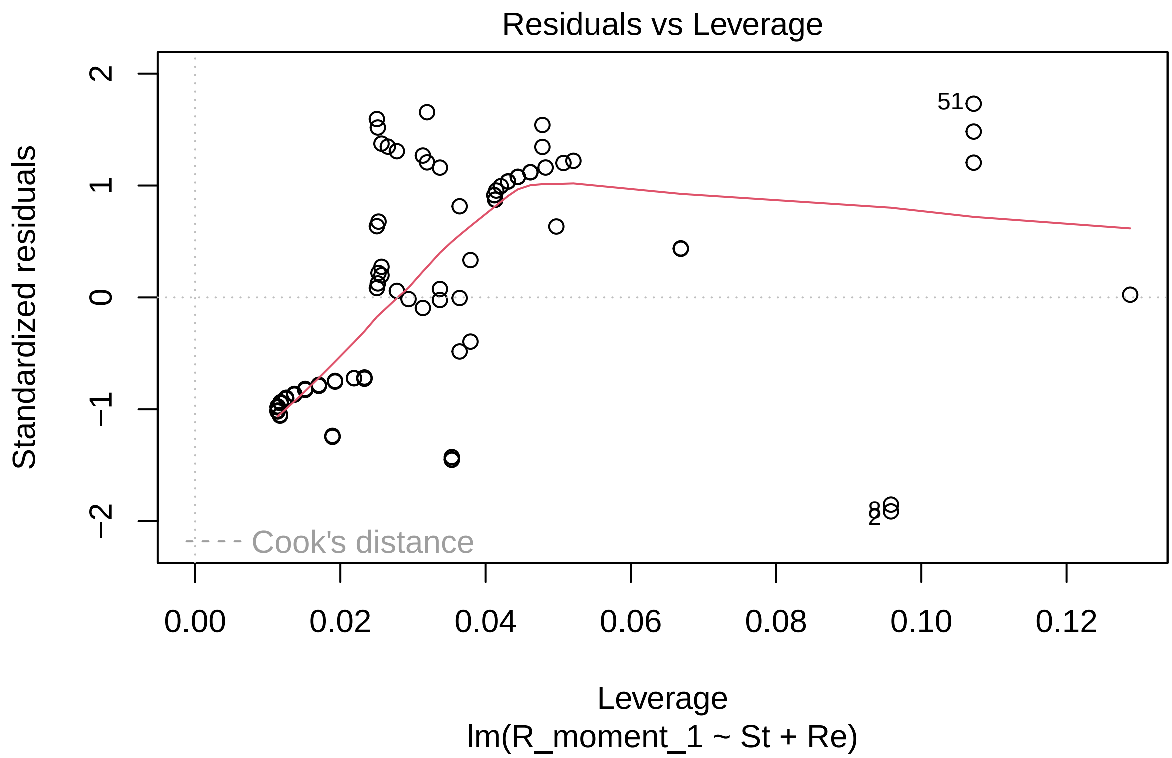
```
## Re          -3.798e-04  3.215e-05 -11.816  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03406 on 86 degrees of freedom
## Multiple R-squared:  0.6364, Adjusted R-squared:  0.628
## F-statistic: 75.26 on 2 and 86 DF,  p-value: < 2.2e-16
```

```
plot(lm_fit_E1)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(R_moment_1 ~ St + Re)
2
1
0
−1
−2
−2
−1
0
1
2
51
2
8


Scale−Location
√|Standardized residuals|
0.0
0.4
0.8
1.2
0.00
0.05
0.10
Fitted values
lm(R_moment_1 ~ St + Re)
51




```r
cve_linear_E1$delta
```

```
## [1] 0.001251509 0.001244488
```

Surprisingly, a very simple linear model with only two out of the three predictors explains about 62% of the variation of the first moment. The Reynolds number coefficient is small and negative, which contradicts physics theory. I believe this is due to the fact that the overwhelming majority of observations had small mean turbulence, so the regression fit a line with negative slope. On average, we just do not often observe turbulence no matter what predictors are used. However, the coefficient on St is slightly larger and positive. I believe this shows that perhaps the most important contributor to increases in the first moment is the size of the particles. In fact, adding interactions or the Fr predictor did not change the R^2 very much, so I believe that St is very important for increasing average turbulence. Nonetheless, there is a clear pattern to the residuals plot. First we underestimate, then overestimate, then underestimate again. This is evidence of a potential nonlinear relationship between the variables and the predictors.

```r
lm_fit_E2 <- lm(R_moment_2_central ~ TFr + Re, data = data_train)
summary(lm_fit_E2)
```

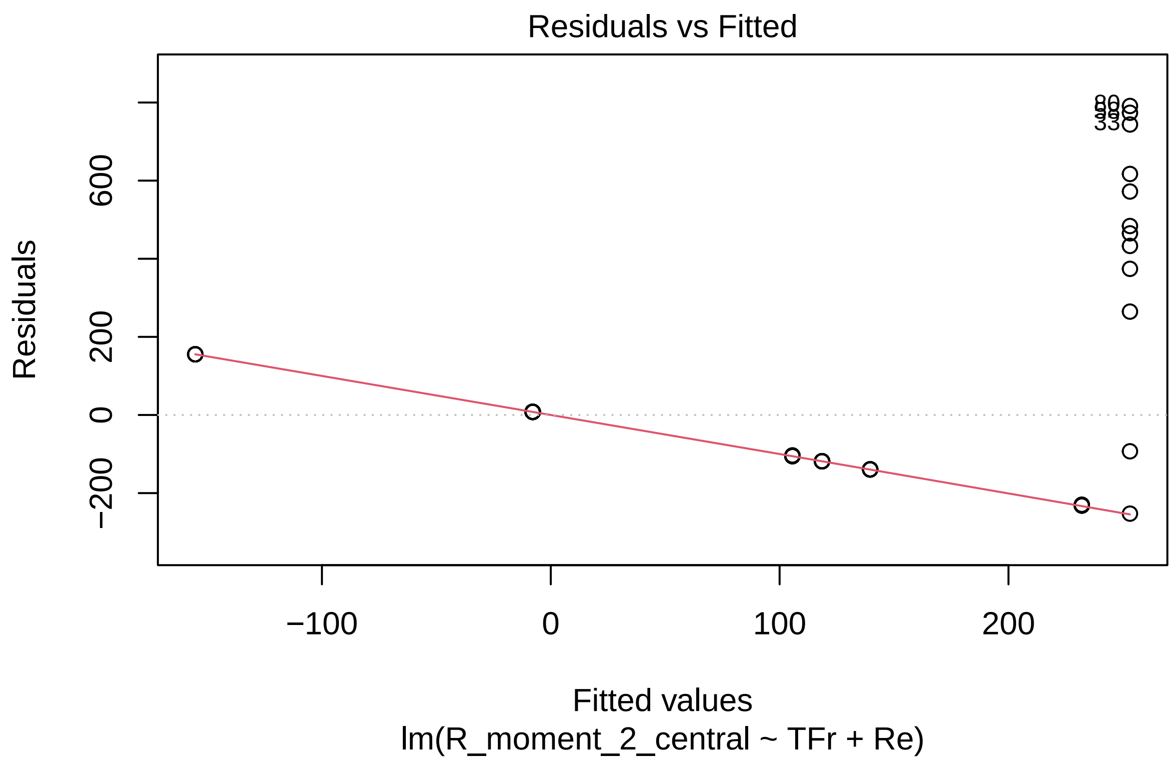
```
##
## Call:
## lm(formula = R_moment_2_central ~ TFr + Re, data = data_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

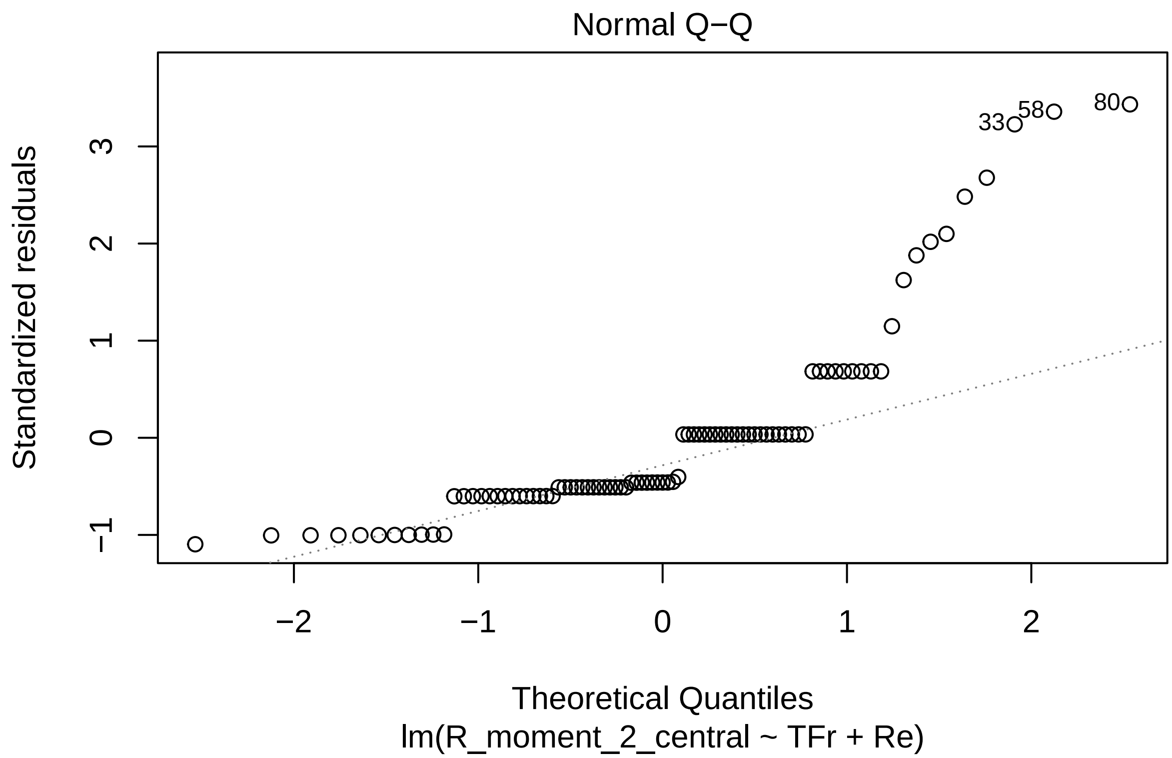
```
## -252.57 -139.16 -104.99    7.98  791.18
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 299.6445    53.6449   5.586 2.68e-07 ***
## TFr         -10.2315     3.8471  -2.660 0.009332 **
## Re           -0.8472     0.2221  -3.815 0.000256 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 234.1 on 86 degrees of freedom
## Multiple R-squared:  0.2189, Adjusted R-squared:  0.2007
## F-statistic: 12.05 on 2 and 86 DF,  p-value: 2.438e-05
```

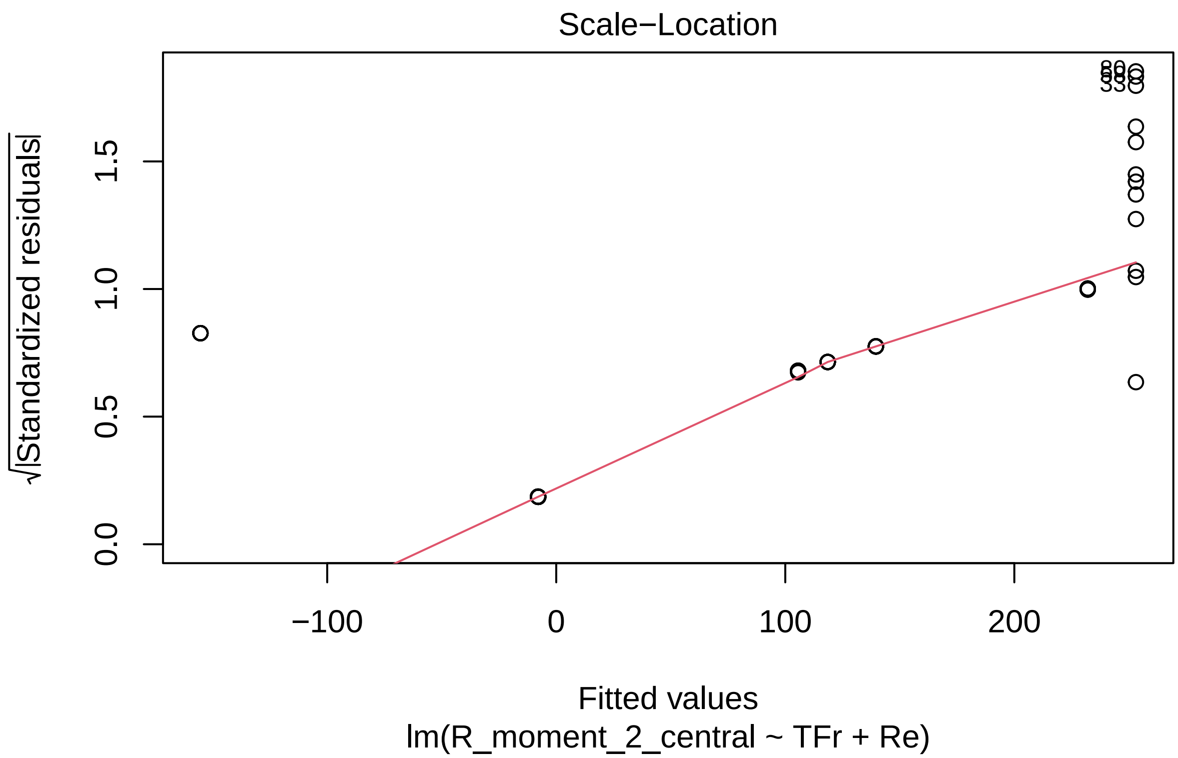
```
cve_linear_E2$delta
```

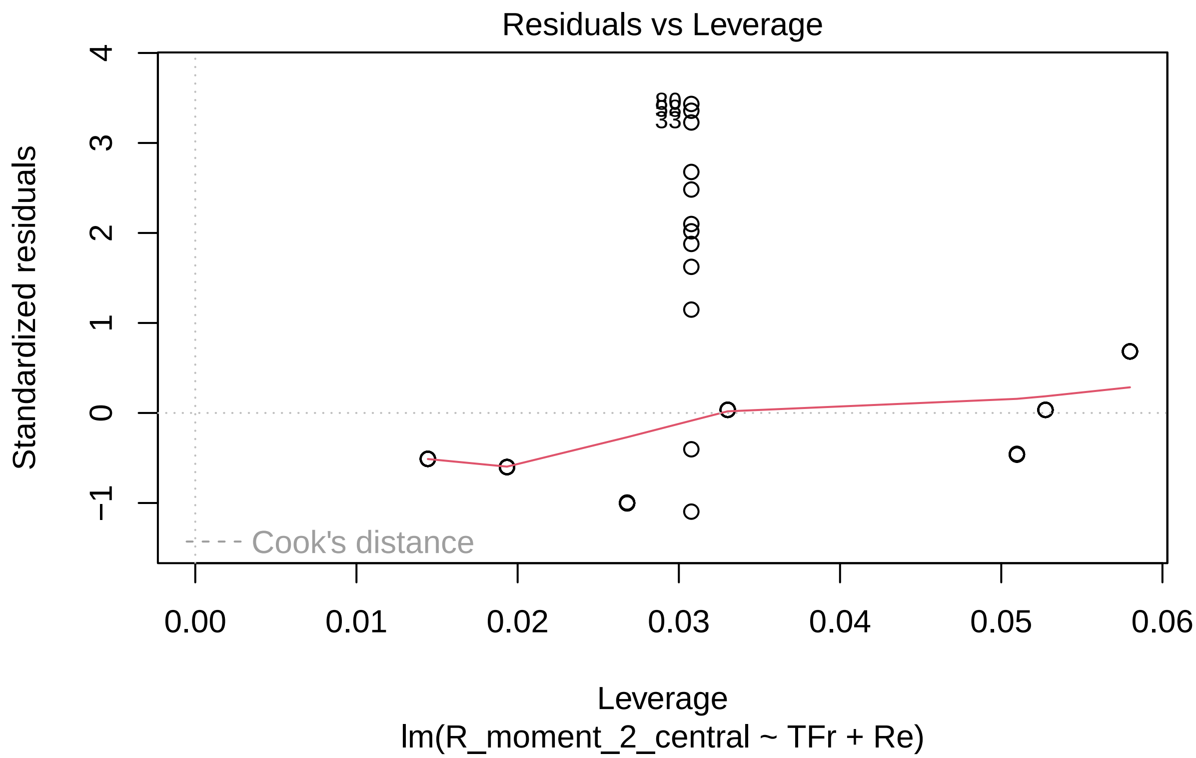
```
## [1] 59566.67 59209.99
```

```
plot(lm_fit_E2)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(R_moment_2_central ~ TFr + Re)
33
58
80
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
−100
0
100
200
Fitted values
lm(R_moment_2_central ~ TFr + Re)



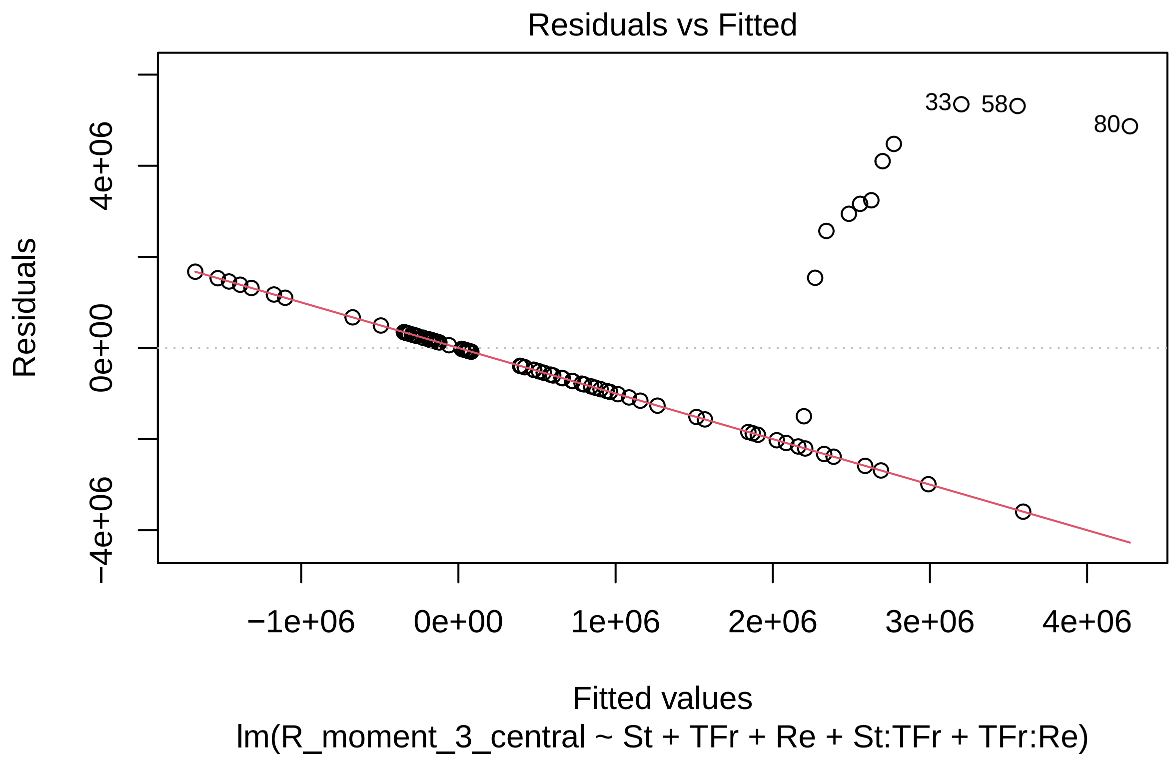
The R^2 is quite low at only about 20%. Generally, I believe this linear model is not very helpful. There is another clear pattern in the residuals plot and the linear fit consistently underestimates when the second moment is large. The truth is definitely closer to a nonlinear relationship.

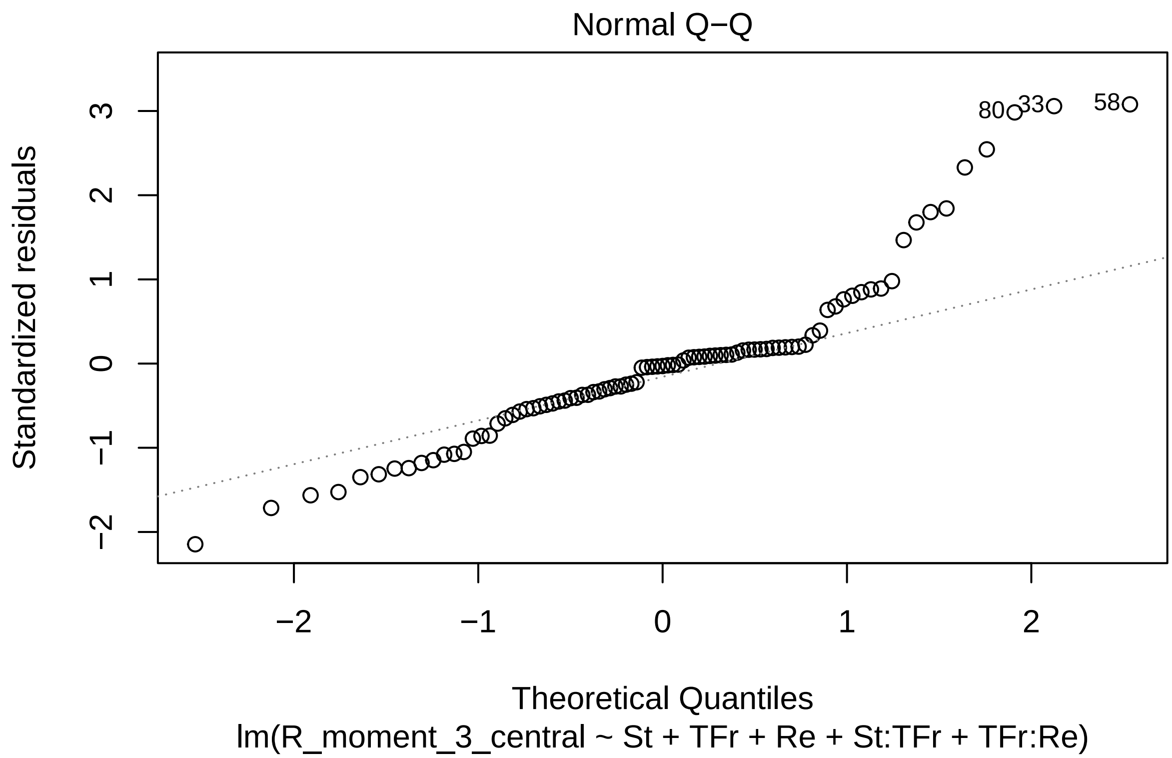
```
lm_fit_E3 <- lm(R_moment_3_central ~ St + TFr + Re + St:TFr + TFr:Re, data = data_train)
summary(lm_fit_E3)
```

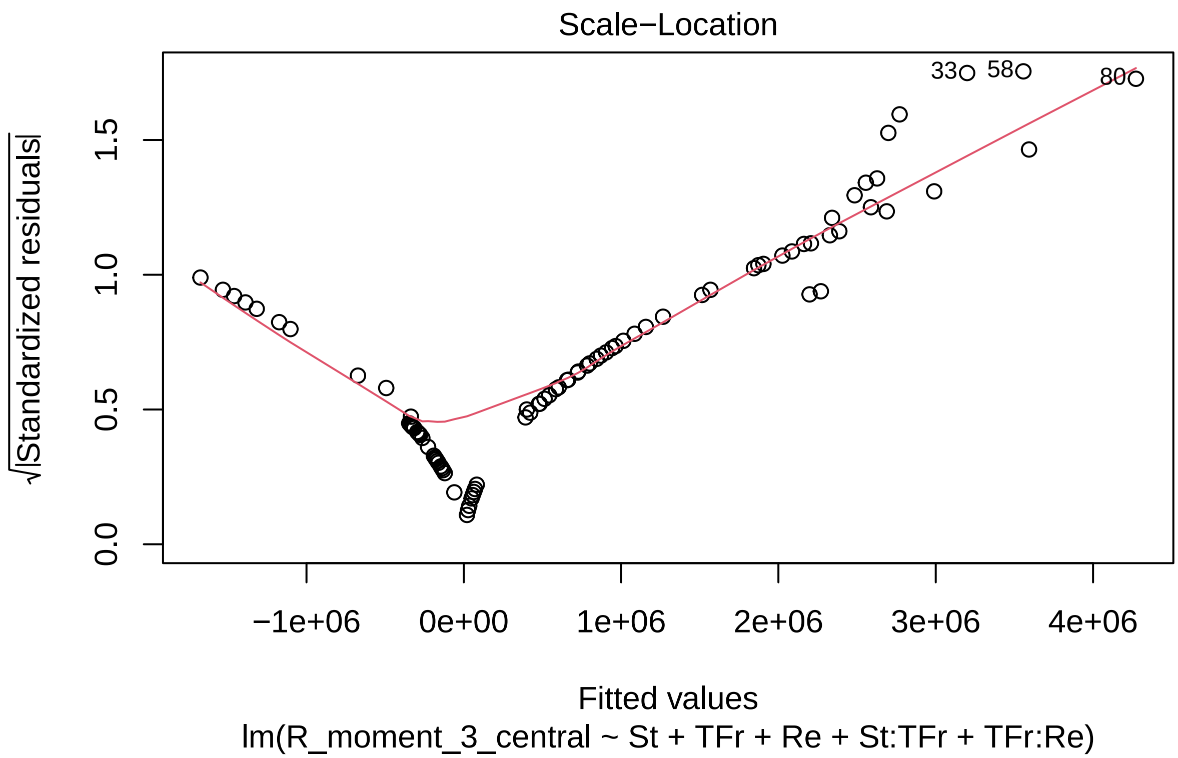
```
##
## Call:
## lm(formula = R_moment_3_central ~ St + TFr + Re + St:TFr + TFr:Re,
##     data = data_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3592944  -904717   -49411   332389  5349989
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2525572.0   493187.5   5.121 1.94e-06 ***
## St           556593.3   258219.6   2.156 0.034018 *
## TFr         -252297.4    72716.5  -3.470 0.000829 ***
## Re            -9805.2     1864.1  -5.260 1.10e-06 ***
## St:TFr       -54689.6    37680.2  -1.451 0.150434
## TFr:Re          953.2      250.9   3.798 0.000276 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

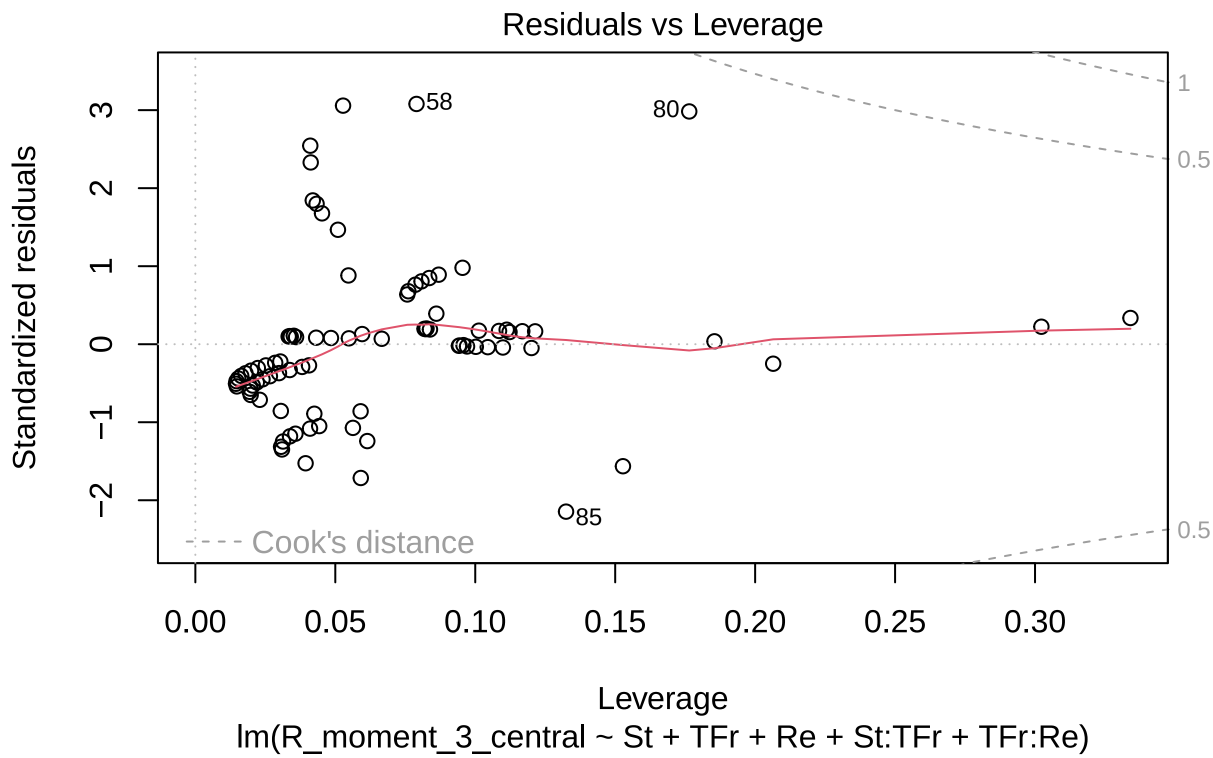
```
##
## Residual standard error: 1798000 on 83 degrees of freedom
## Multiple R-squared:  0.3603, Adjusted R-squared:  0.3218
## F-statistic:  9.35 on 5 and 83 DF,  p-value: 4.292e-07
```

```
plot(lm_fit_E3)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(R_moment_3_central ~ St + TFr + Re + St:TFr + TFr:Re)
80
33
58
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
−1e+06
0e+00
1e+06
2e+06
3e+06
4e+06
Fitted values
lm(R_moment_3_central ~ St + TFr + Re + St:TFr + TFr:Re)
33
58
80




```
cve_linear_interactions_E3$delta
```

```
## [1] 3.442602e+12 3.418975e+12
```

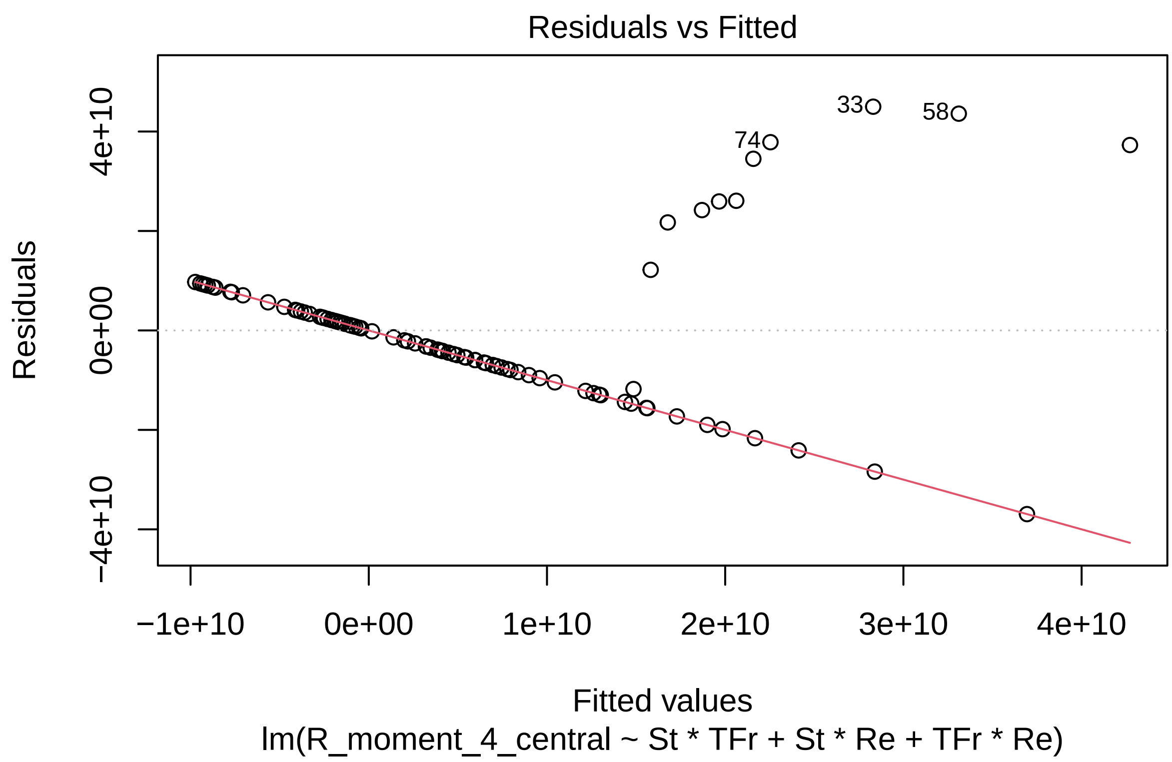
The R^2 is still not great at only about 32%. However, the interaction terms give important theoretical insights that are more in line with the limited theory that we know. The coefficient on TFr:Re is positive, which means that even though the coefficients on Re and TFr alone are negative, we can infer that at high enough levels of TFr, the effect of Re will actually be positive (since the interaction term means that for a given TFr, the coefficient on Re is (-9805.2 + 953.2 * TFr)). Similarly, the effect of TFr will be positive at high enough levels of Re. Thus, increasing rightward skewness of the probability density functions with Re or TFr seems to occur only for a combination of high values of TFr and Re. Otherwise, the main positive coefficient is St. Again, we see that the size of the particles has a particularly straightforward effect on turbulence. It increases the first moment and the right skewness of the PDF.

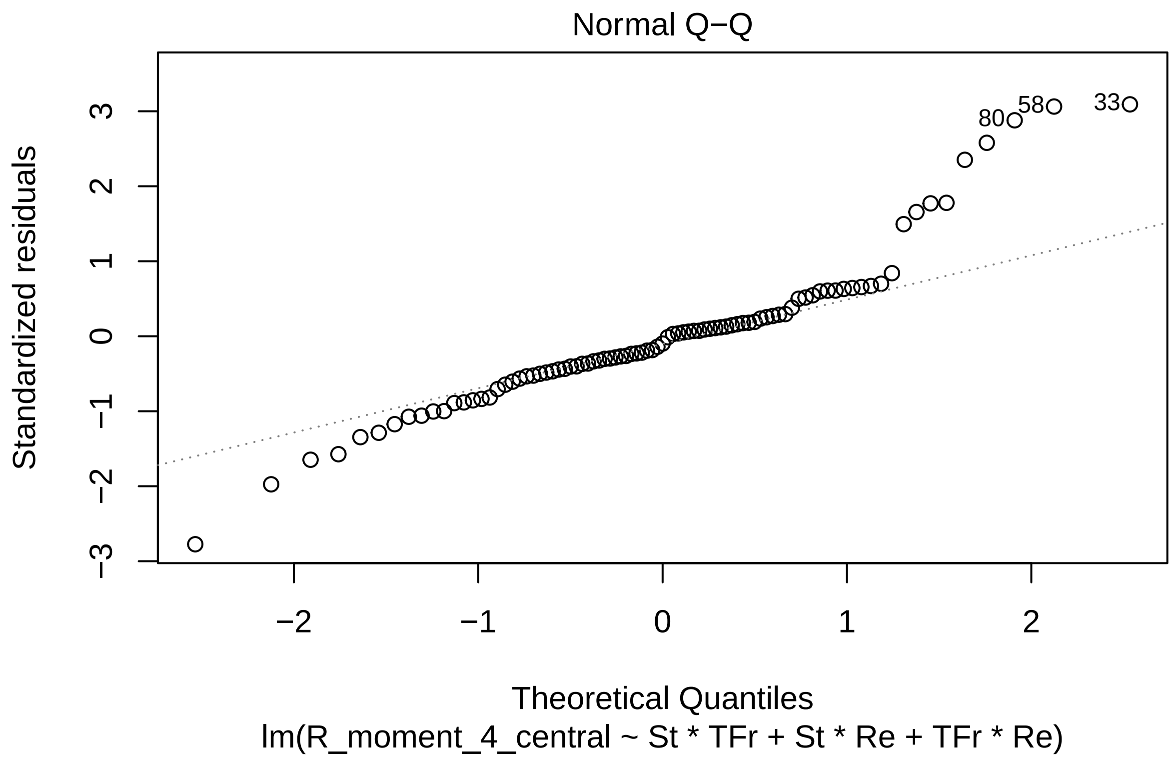
```
lm_fit_E4 <- lm(R_moment_4_central ~ St * TFr + St * Re + TFr * Re, data = data_train)
summary(lm_fit_E4)
```

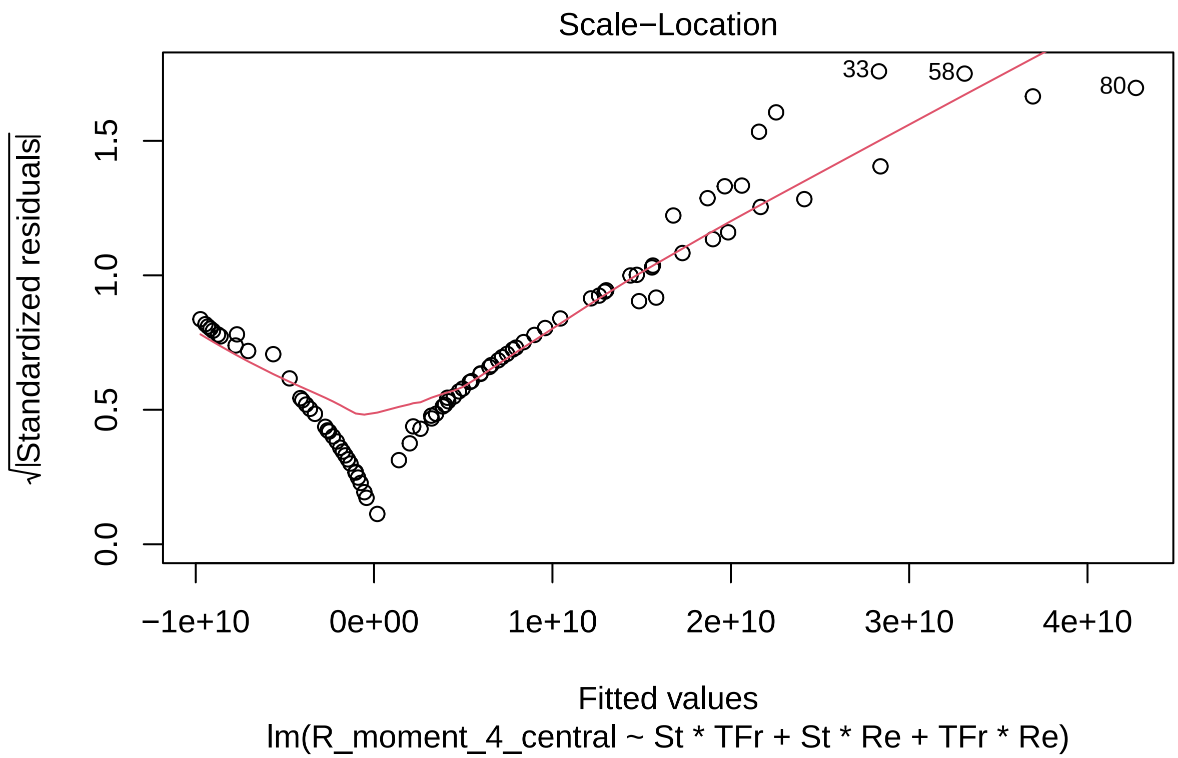
```
##
## Call:
## lm(formula = R_moment_4_central ~ St * TFr + St * Re + TFr *
##     Re, data = data_train)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
```

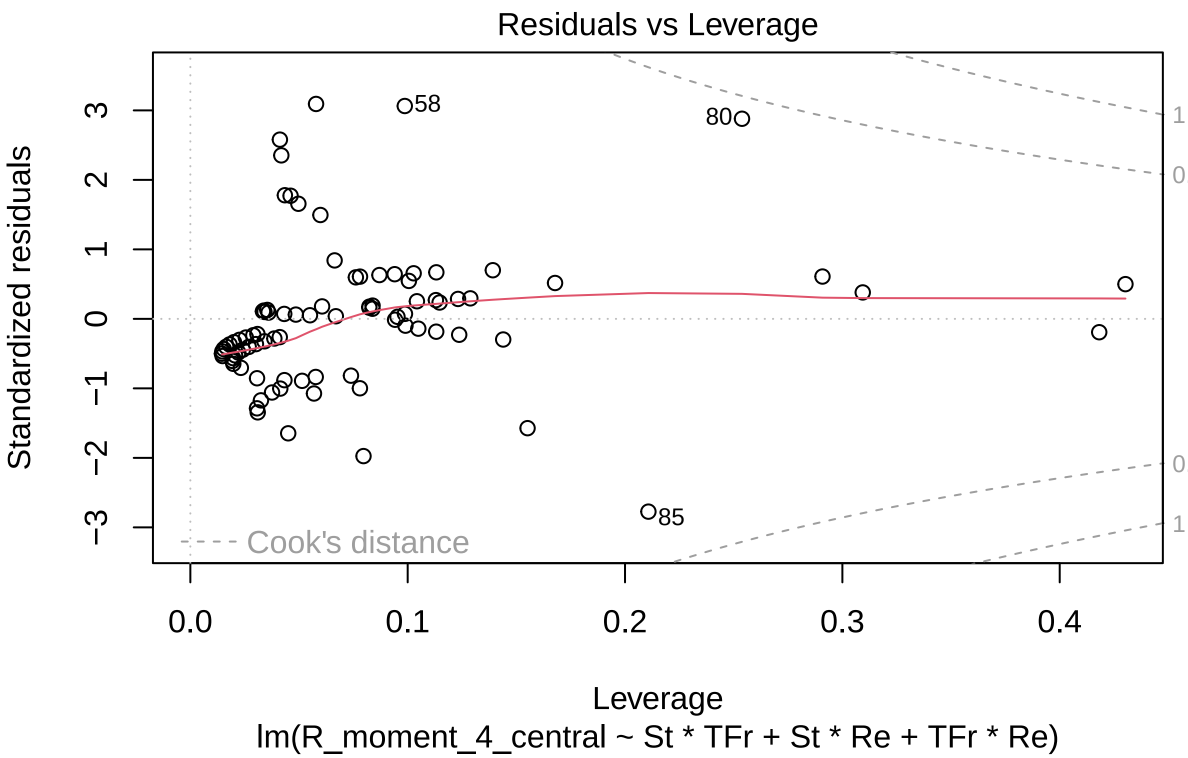
```
## -3.694e+10 -7.457e+09 -1.392e+09  4.131e+09  4.499e+10
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.528e+10  5.349e+09   2.857 0.005418 **
## St           1.050e+10  4.256e+09   2.467 0.015707 *
## TFr         -1.915e+09  6.113e+08  -3.133 0.002401 **
## Re          -5.598e+07  2.293e+07  -2.442 0.016773 *
## St:TFr      -5.176e+08  3.144e+08  -1.646 0.103499
## St:Re       -2.662e+07  1.816e+07  -1.466 0.146598
## TFr:Re       7.303e+06  2.125e+06   3.438 0.000924 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.499e+10 on 82 degrees of freedom
## Multiple R-squared:  0.3735, Adjusted R-squared:  0.3277
## F-statistic: 8.147 on 6 and 82 DF,  p-value: 6.439e-07
```

```
plot(lm_fit_E4)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(R_moment_4_central ~ St * TFr + St * Re + TFr * Re)
80
58
33
−3
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
33
58
80
−1e+10
0e+00
1e+10
2e+10
3e+10
4e+10
Fitted values
lm(R_moment_4_central ~ St * TFr + St * Re + TFr * Re)



Our analysis of the best linear model for the fourth central moment is quite similar to that of the third central moment. St is the biggest positive driver of kurtosis in the PDF. TFr and Re individually are negative, but they have a positive interaction coefficient.

Complex Model

Lets First start start by evaluating a gam model to see if there is a complex relationship between our response variables and predictor variables. For this section of the project, I will be only employing the simpler lm function to keep all the models in the same format.

```
gam1<-lm(R_moment_1~ns(TFr,2)+ns(Re,2)+ns(St,3),data=data_train)
gam2<-lm(R_moment_2~ns(TFr,2)+ns(Re,2)+ns(St,3),data=data_train)
gam3<-lm(R_moment_3~ns(TFr,2)+ns(Re,2)+ns(St,3),data=data_train)
gam4<-lm(R_moment_4~ns(TFr,2)+ns(Re,2)+ns(St,3),data=data_train)
print(summary(gam1))
```

```
##
## Call:
## lm(formula = R_moment_1 ~ ns(TFr, 2) + ns(Re, 2) + ns(St, 3),
##     data = data_train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.036044 -0.007761  0.000876  0.009547  0.039340
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.103397   0.005175  19.979  < 2e-16 ***
## ns(TFr, 2)1 -0.028702   0.013984  -2.052   0.0434 *
## ns(TFr, 2)2 -0.003101   0.004294  -0.722   0.4722
## ns(Re, 2)1  -0.218972   0.006966 -31.434  < 2e-16 ***
## ns(Re, 2)2  -0.051078   0.004516 -11.311  < 2e-16 ***
## ns(St, 3)1   0.012423   0.008158   1.523   0.1317
## ns(St, 3)2   0.040473   0.009599   4.216 6.42e-05 ***
## ns(St, 3)3   0.033807   0.006324   5.346 8.10e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01536 on 81 degrees of freedom
## Multiple R-squared:  0.9304, Adjusted R-squared:  0.9244
## F-statistic: 154.7 on 7 and 81 DF,  p-value: < 2.2e-16
```

```
print(summary(gam2))
```

```
##
## Call:
## lm(formula = R_moment_2 ~ ns(TFr, 2) + ns(Re, 2) + ns(St, 3),
##     data = data_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -328.27 -163.57  -25.44  105.75  595.24
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   328.80      67.37   4.881 5.22e-06 ***
## ns(TFr, 2)1  -935.18     182.04  -5.137 1.89e-06 ***
## ns(TFr, 2)2    35.10      55.89   0.628  0.53171
## ns(Re, 2)1   -548.96      90.68  -6.054 4.20e-08 ***
## ns(Re, 2)2   -173.57      58.78  -2.953  0.00412 **
## ns(St, 3)1     86.24     106.20   0.812  0.41916
## ns(St, 3)2    214.67     124.96   1.718  0.08963 .
## ns(St, 3)3     69.71      82.32   0.847  0.39958
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 199.9 on 81 degrees of freedom
## Multiple R-squared:  0.4634, Adjusted R-squared:  0.417
## F-statistic: 9.992 on 7 and 81 DF,  p-value: 6.159e-09
```

```
print(summary(gam3))
```

```
##
## Call:
## lm(formula = R_moment_3 ~ ns(TFr, 2) + ns(Re, 2) + ns(St, 3),
##     data = data_train)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2630267 -1310948  -161527   852632  5383248
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2630290     569154   4.621 1.42e-05 ***
## ns(TFr, 2)1 -7633365    1537912  -4.963 3.77e-06 ***
## ns(TFr, 2)2   290780     472190   0.616   0.5397
## ns(Re, 2)1  -4471187     766090  -5.836 1.06e-07 ***
## ns(Re, 2)2  -1413605     496608  -2.847   0.0056 **
## ns(St, 3)1    816362     897205   0.910   0.3656
## ns(St, 3)2   1896317    1055663   1.796   0.0762 .
## ns(St, 3)3    707565     695437   1.017   0.3120
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1689000 on 81 degrees of freedom
## Multiple R-squared:  0.4491, Adjusted R-squared:  0.4015
## F-statistic: 9.433 on 7 and 81 DF,  p-value: 1.659e-08
```

```
print(summary(gam4))
```

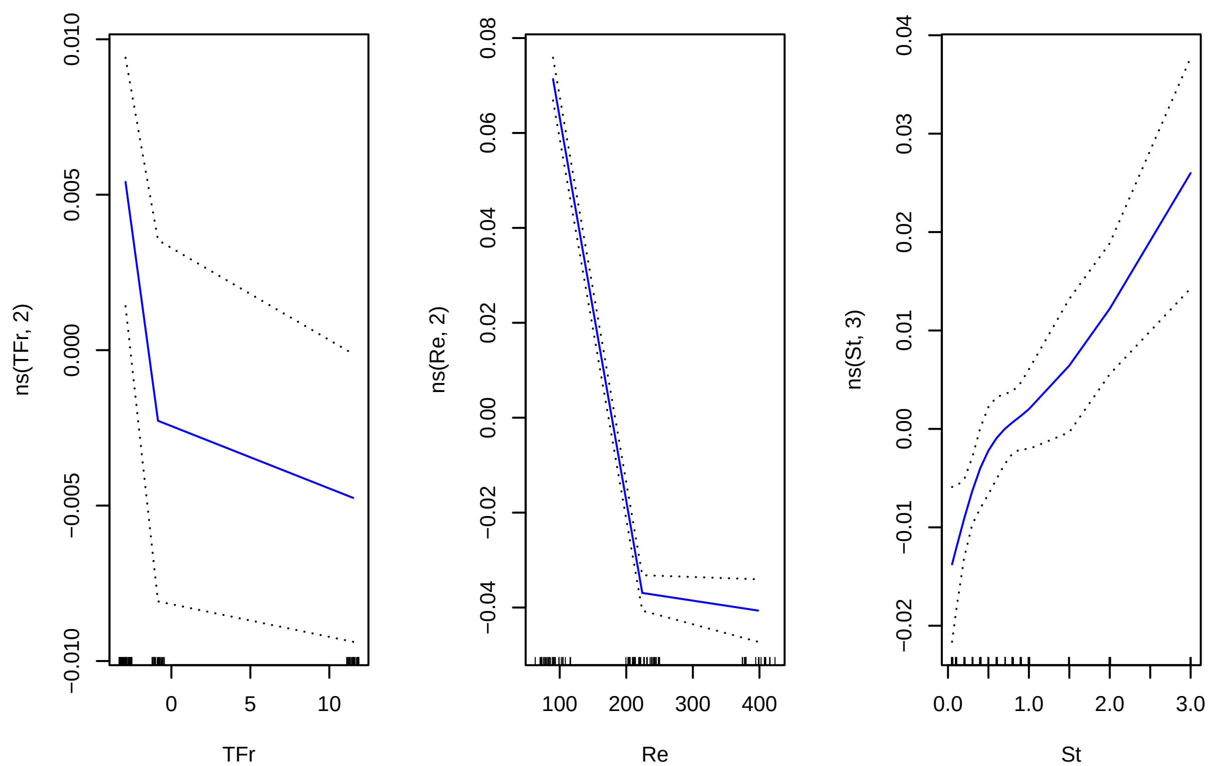
```
##
## Call:
## lm(formula = R_moment_4 ~ ns(TFr, 2) + ns(Re, 2) + ns(St, 3),
##     data = data_train)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.134e+10 -1.051e+10 -1.073e+09  6.815e+09  4.835e+10
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.134e+10  4.817e+09   4.431 2.91e-05 ***
## ns(TFr, 2)1 -6.266e+10  1.302e+10  -4.814 6.77e-06 ***
## ns(TFr, 2)2  2.406e+09  3.996e+09   0.602  0.54885
## ns(Re, 2)1  -3.670e+10  6.483e+09  -5.660 2.21e-07 ***
## ns(Re, 2)2  -1.160e+10  4.203e+09  -2.759  0.00716 **
## ns(St, 3)1   7.560e+09  7.593e+09   0.996  0.32239
## ns(St, 3)2   1.642e+10  8.934e+09   1.838  0.06969 .
## ns(St, 3)3   6.915e+09  5.886e+09   1.175  0.24347
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.429e+10 on 81 degrees of freedom
## Multiple R-squared:  0.4374, Adjusted R-squared:  0.3888
## F-statistic: 8.996 on 7 and 81 DF,  p-value: 3.651e-08
```

As we can see, a Gam Model performs better with all the moments than linear regression. Nevertheless, as we can see the model only performs adequately with the first moment with an adjusted r square of 0.9244 and an RSS of 0.01536 compared to an average of 0.4 for the other models. This likely is due to the lack of interaction factors in our model, which appear to affect the second, third and fourth moment more than the first. Thus, we might need to add some interaction values to our polynomial model.

Note: 2 degrees were chosen due to the number of unique values of in our data. Only 3 degrees were chosen for St since it has multiple unique values.

Plots Since the first GAM performed particulary well, it might be worthwhile to explore the relationship between our response varible and predictor varibles using plots.

```
par(mfrow=c(1,3))
gam_one<- gam(R_moment_1~ns(TFr,2)+ns(Re,2)+ns(St,3),data=data_train)
plot(gam_one, se = TRUE, col = "blue")
```



As we can see from the plots, the first moment appears to experience a steep drop for both TFr and Re from the first to the second observation. The decrease becomes much less significant from the second observation to the third observation for both Tfr and Re. The relationship between St and the first moment appear to be roughly linear and increasing.

Best Degree Model We can utilize a sequential stepwise selection method with a full model. Our full model would utilize the max number of interactions with a polynomial degree of 2 for TFr and Re (Max number of degrees based on unique values). We will use one for St since throughout our linear models and GAM models, it appears that the relationship between the St and the moments is approximately linear.

```
train.control <- trainControl(method = "LOOCV")
step.model.one <- train(R_moment_1~I(TFr^2)*I(Re^2)*St+St*Re*TFr, data = data_train,
                        method = "leapSeq",
                        tuneGrid = data.frame(nvmax =1:13),
                        trControl = train.control
```

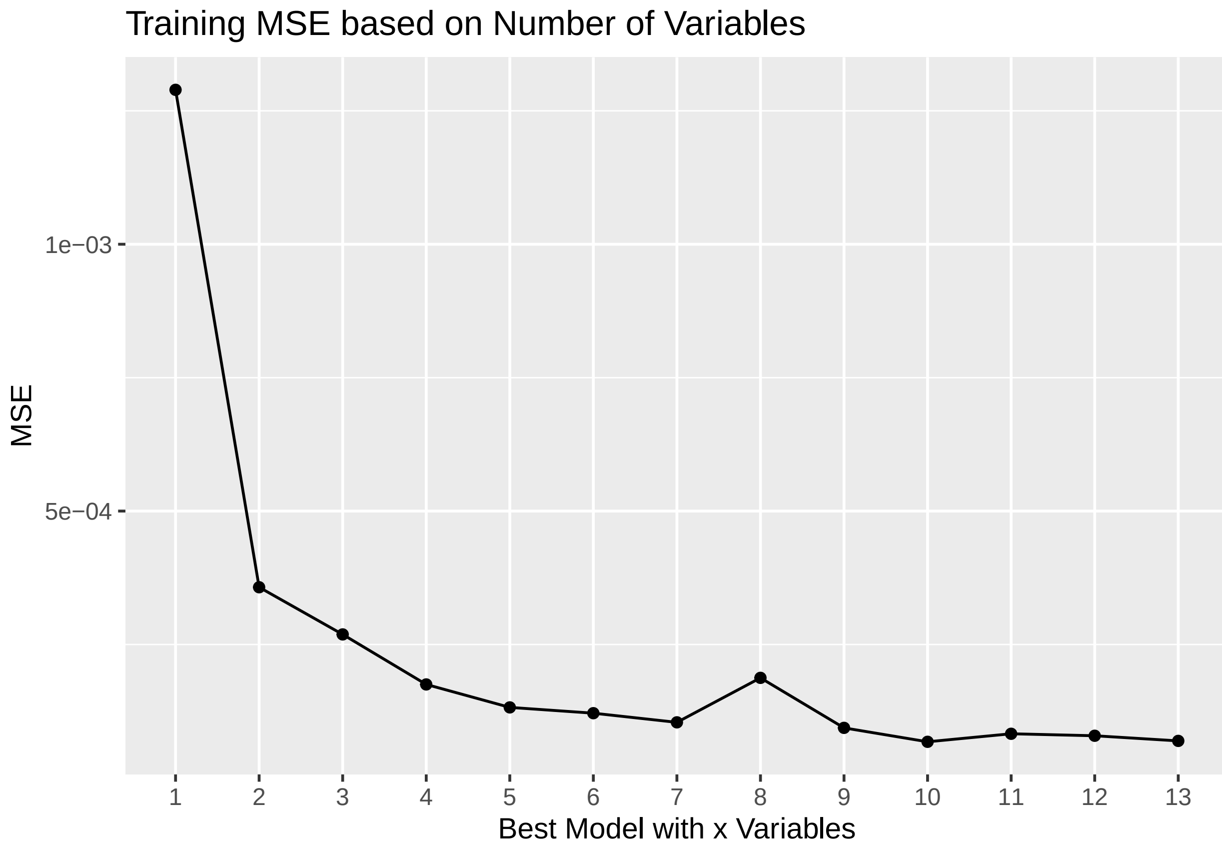
```r
                    )
step.model.one$results
```

```
##    nvmax        RMSE  Rsquared         MAE
## 1      1 0.035907108 0.5821314 0.032637091
## 2      2 0.018902099 0.8841424 0.010019540
## 3      3 0.016397990 0.9128001 0.012347020
## 4      4 0.013233933 0.9432366 0.009701428
## 5      5 0.011494721 0.9571816 0.006110876
## 6      6 0.011012601 0.9607081 0.008001465
## 7      7 0.010203352 0.9662665 0.007147665
## 8      8 0.013691329 0.9392635 0.010374143
## 9      9 0.009683652 0.9696395 0.007448496
## 10    10 0.008224892 0.9781803 0.005365646
## 11    11 0.009094052 0.9733865 0.006022789
## 12    12 0.008886783 0.9745549 0.005927554
## 13    13 0.008328647 0.9776521 0.005553642
```

```r
coef(step.model.one$finalModel, 5)
```

```
##   (Intercept)       I(Re^2)            St            Re    I(Re^2):St
##  1.822255e-01  1.993194e-06  6.143005e-02 -1.250385e-03  6.666005e-07
##         St:Re
## -4.195074e-04
```

```r
data1<-data.frame(step.model.one$results$nvmax, step.model.one$results$RMSE)
ggplot(data1,aes(x = step.model.one.results.nvmax, y =(step.model.one.results.RMSE)^2))+
         geom_line()+
         geom_point()+
  scale_x_continuous(breaks = 1:13, minor_breaks = NULL) +
   labs(title = "Training MSE based on Number of Variables ",
        x="Best Model with x Variables", y="MSE")
```



As we can see from the plot, there appears to be a decrease in MSE until about the 5 degrees best model. Then the MSE increases again until about the model with 9 variables, where it decreases again. Since, for the first moment, we are focusing more on inference and the error appears to be negligible between the fifth model and later models, we will choose the fifth model as our best model.

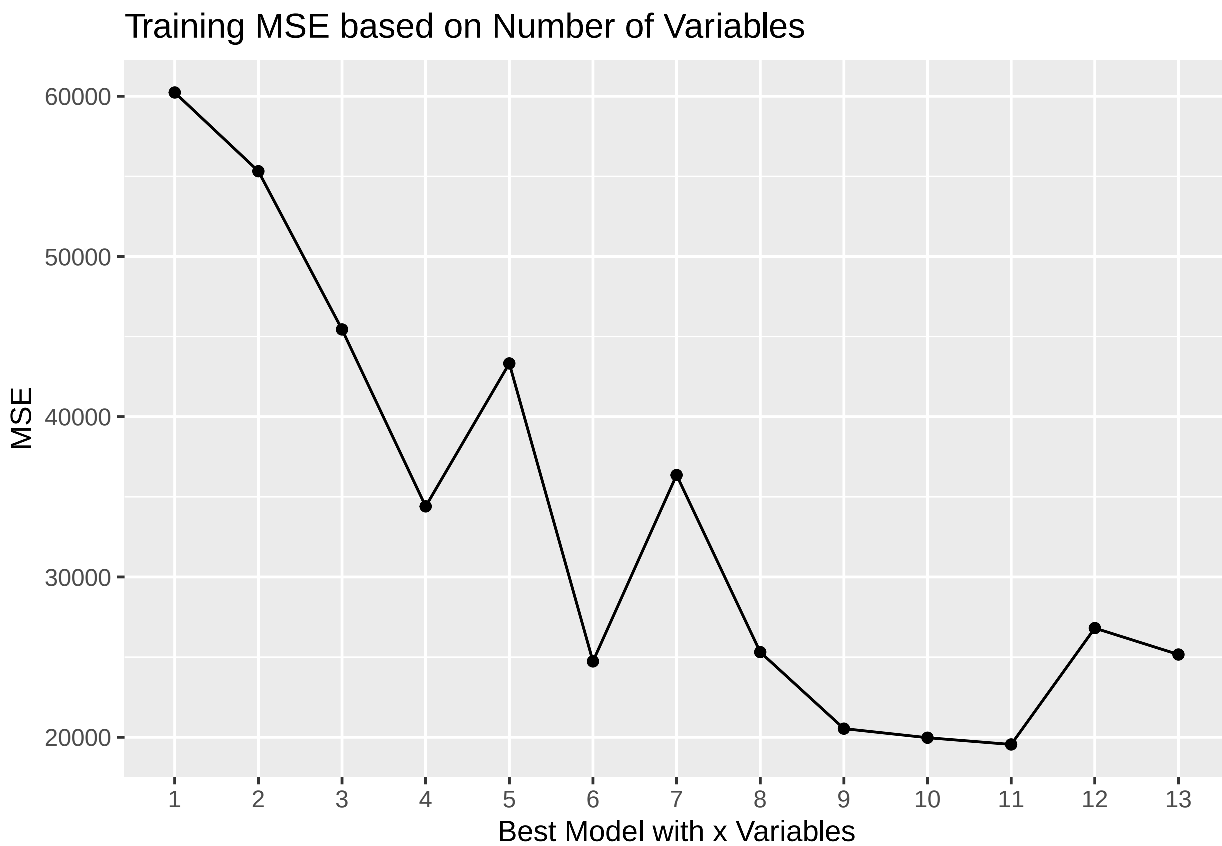
Next, we are going to repeat the same process for all the other moments with a focus on prediction instead of inference.

```
step.model.two <- train(R_moment_2~I(TFr^2)*I(Re^2)*St+St*Re*TFr, data = data_train,
                    method = "leapSeq",
                    tuneGrid = data.frame(nvmax =1:13),
                    trControl = train.control
                    )
step.model.two$results
```

```
##    nvmax     RMSE  Rsquared      MAE
## 1      1 245.4204 0.1146493 170.5543
## 2      2 235.1982 0.1867807 125.3575
## 3      3 213.1774 0.3317056 156.7218
## 4      4 185.4832 0.4934326 128.1483
## 5      5 208.1486 0.3656905 172.2743
## 6      6 157.2638 0.6360246 112.6197
## 7      7 190.6828 0.4694553 152.2684
## 8      8 159.0946 0.6303544 119.3594
## 9      9 143.3013 0.6988549 110.4690
## 10    10 141.3121 0.7097159 110.7573
## 11    11 139.8139 0.7157430 105.2612
```

```
## 12    12 163.7361 0.6403635 115.7253
## 13    13 158.6199 0.6606477 117.0337
```

```
data2<-data.frame(step.model.two$results$nvmax, step.model.two$results$RMSE)
ggplot(data2,aes(x = step.model.two.results.nvmax, y =(step.model.two.results.RMSE)^2))+
         geom_line()+
         geom_point()+
  scale_x_continuous(breaks = 1:13, minor_breaks = NULL) +
   labs(title = "Training MSE based on Number of Variables ",
        x="Best Model with x Variables", y="MSE")
```



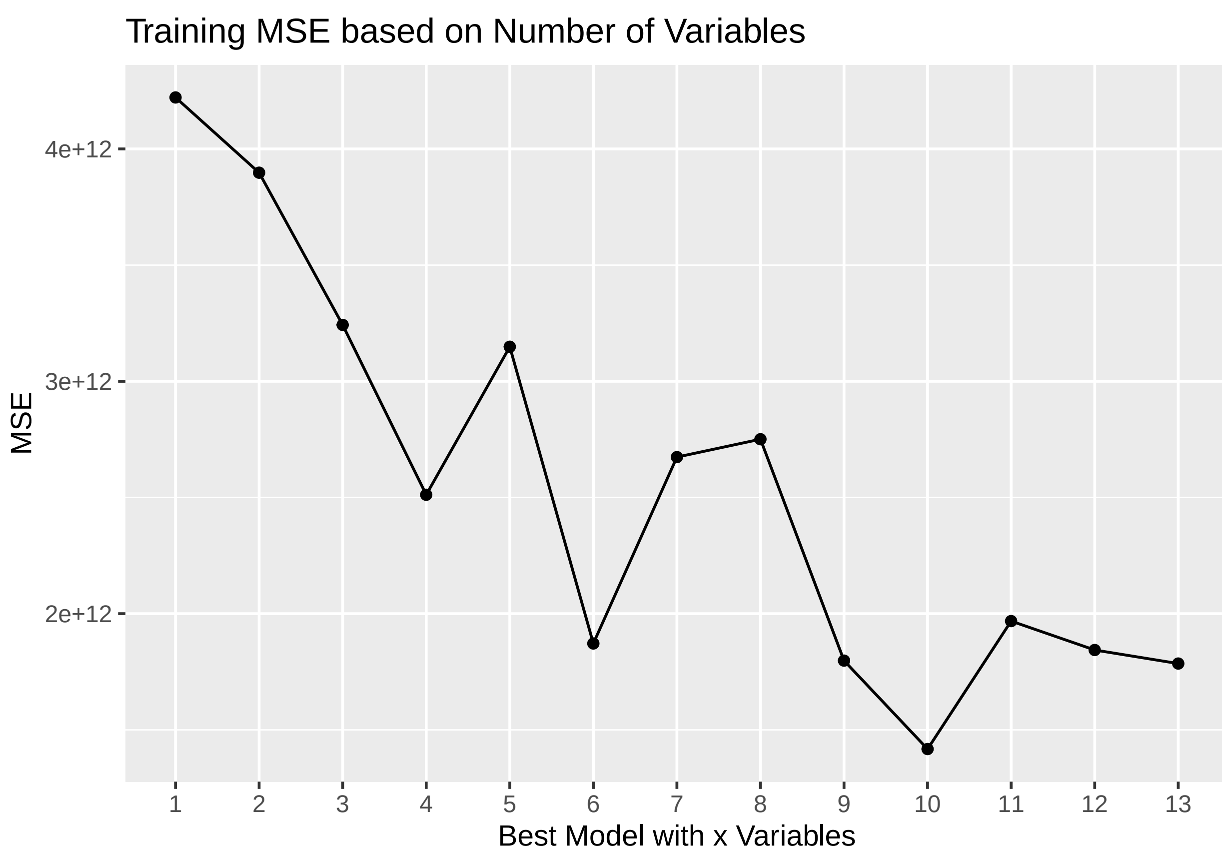
As we can see from the plot, the model does best is the one with 12 variables. However, this model is most likely not very understandable since it does include a variety of interactions including the three way interaction. So the model with 9 variables might be better for some inference. [Francis: I would prefer to go with 6 because even that is a lot of variables already.] Nevertheless, it does appear that the higher the moment the harder it is to predict.

```
step.model.three <- train(R_moment_3~I(TFr^2)*I(Re^2)*St+St*Re*TFr, data = data_train,
                    method = "leapSeq",
                    tuneGrid = data.frame(nvmax =1:13),
                    trControl = train.control
                    )
step.model.three$results
```

```
##    nvmax    RMSE  Rsquared       MAE
## 1      1 2054619 0.1076165 1407528.9
```

```
## 2       2 1974257 0.1759897 1038479.2
## 3       3 1800729 0.3142725 1295715.6
## 4       4 1584833 0.4682023 1067444.4
## 5       5 1774453 0.3394066 1436102.4
## 6       6 1368088 0.6039419  946062.3
## 7       7 1635166 0.4375141 1267899.5
## 8       8 1658524 0.4474426 1273711.8
## 9       9 1340900 0.6198440  952179.2
## 10     10 1190552 0.7038748  921391.3
## 11     11 1402872 0.6227703 1043704.4
## 12     12 1357809 0.6486801  911926.4
## 13     13 1336150 0.6570828  966935.1
```

```
data2<-data.frame(step.model.three$results$nvmax, step.model.three$results$RMSE)
ggplot(data2,aes(x = step.model.three.results.nvmax, y =(step.model.three.results.RMSE)^2))+
         geom_line()+
         geom_point()+
  scale_x_continuous(breaks = 1:13, minor_breaks = NULL) +
   labs(title = "Training MSE based on Number of Variables ",
        x="Best Model with x Variables", y="MSE")
```

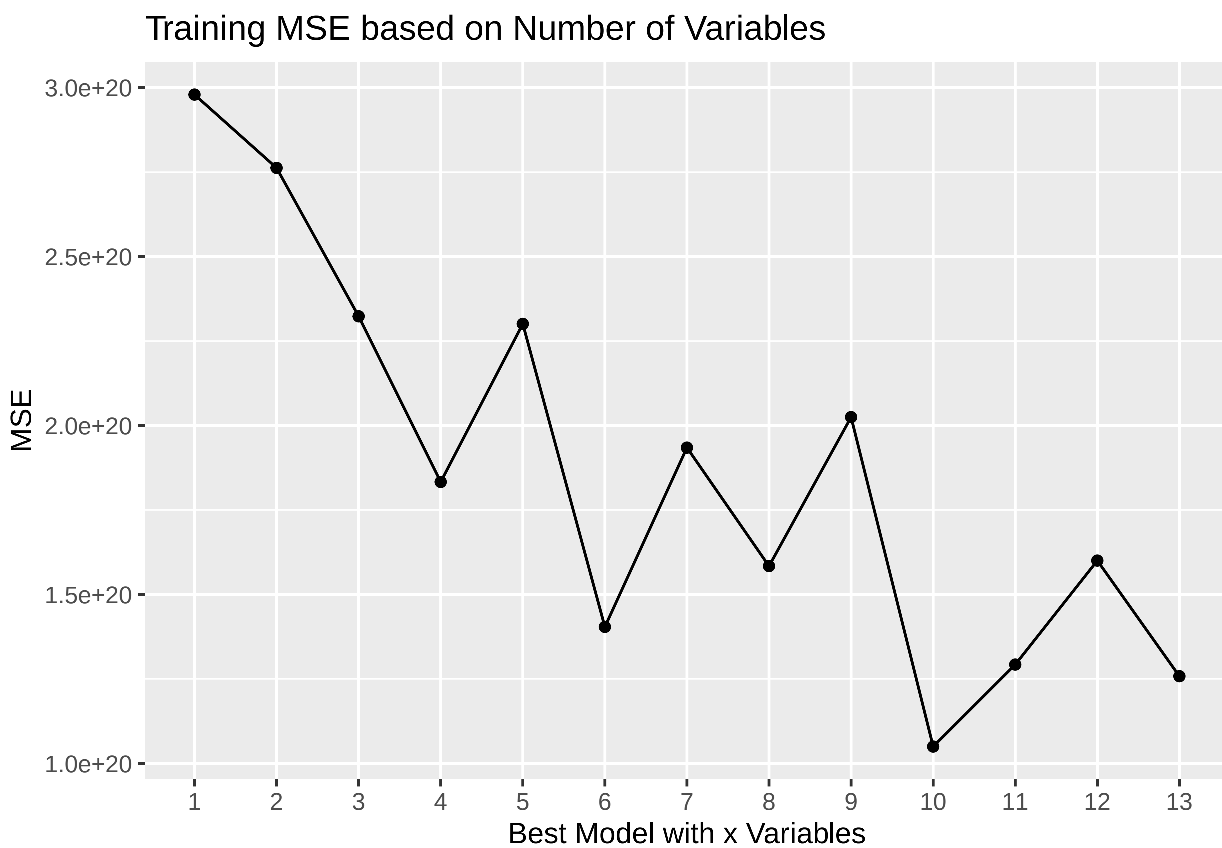


As we can see, the complex model (the almost full model with 12 variables) appears to be doing significantly best. However, since the errors are so big, it might be worth exploring the difference in error between including 12 variables and 8 variables in our final model. [Francis: 6 looks fine again]

```r
step.model.four<- train(R_moment_4~I(TFr^2)*I(Re^2)*St+St*Re*TFr, data = data_train,
                    method = "leapSeq",
                    tuneGrid = data.frame(nvmax =1:13),
                    trControl = train.control
                    )
step.model.four$results
```

```
##    nvmax        RMSE  Rsquared         MAE
## 1      1 17260728601 0.1021434 11634306167
## 2      2 16620293460 0.1674813  8600375860
## 3      3 15242019369 0.2996575 10716341906
## 4      4 13538858265 0.4467839  8919963437
## 5      5 15168532775 0.3152105 12026775719
## 6      6 11849332824 0.5765145  7959608150
## 7      7 13908928977 0.4202629 10563572015
## 8      8 12585391177 0.5291383  8609824224
## 9      9 14229486417 0.4277501 10536571593
## 10    10 10246541749 0.6899035  7729626426
## 11    11 11369201862 0.6441956  7737004670
## 12    12 12650137730 0.5979081  8901167434
## 13    13 11216554941 0.6581971  7965270940
```

```r
data2<-data.frame(step.model.four$results$nvmax, step.model.four$results$RMSE)
ggplot(data2,aes(x = step.model.four.results.nvmax, y =(step.model.four.results.RMSE)^2))+
         geom_line()+
         geom_point()+
  scale_x_continuous(breaks = 1:13, minor_breaks = NULL) +
   labs(title = "Training MSE based on Number of Variables ",
        x="Best Model with x Variables", y="MSE")
```



Interestingly, for the fourth moment, the full model appears to do the best. However, the model with 8 variables might of interest to explore since it might be better for inference while not sacrificing a ton of accuracy. [Francis: That's a good idea. Although for kurtosis (the fourth moment) we really aren't learning much about the distribution to begin with so there is no harm in aiming for good prediction and having a flexible model with 6 or 10 variables. I think it might be worth trying 4 variables for the other models 2 & 3 instead, so we can do more single variable plots.]

Results (Final Models)

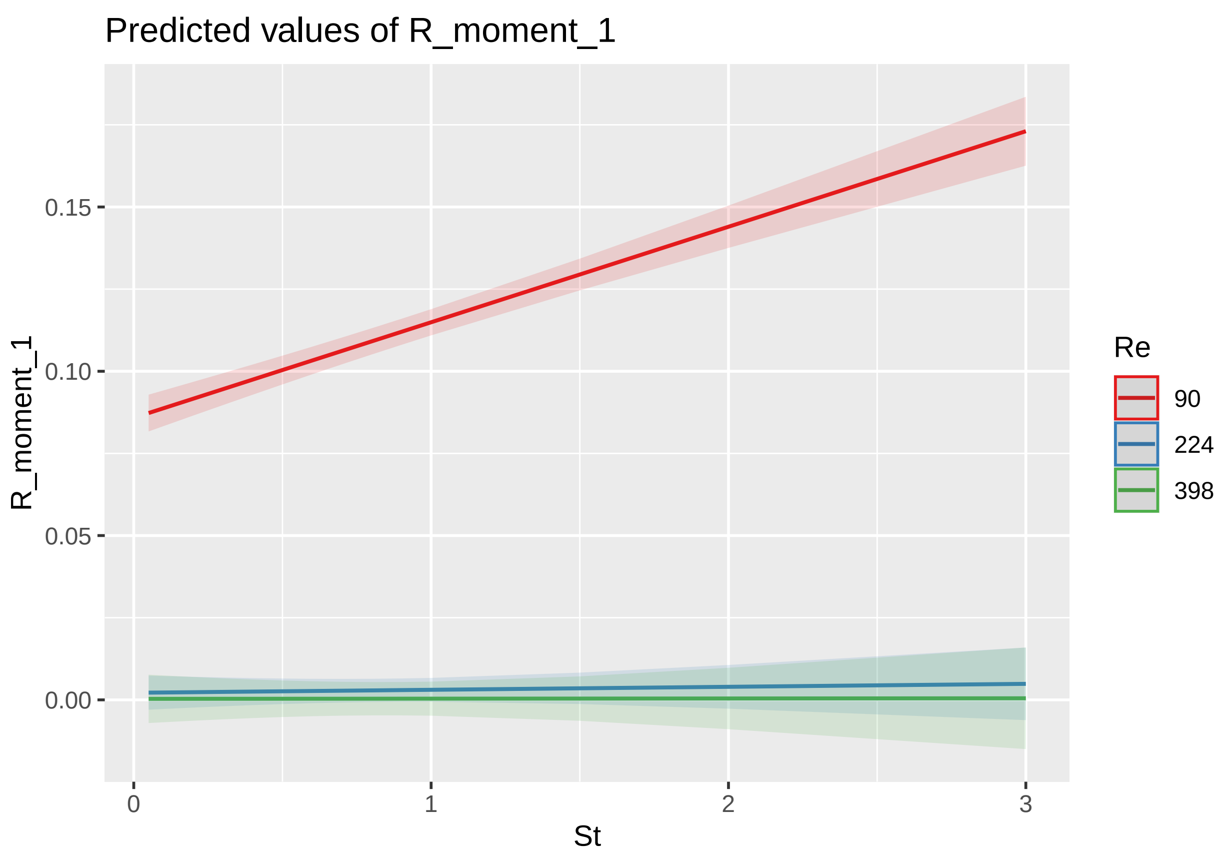
```
model_1<-lm(R_moment_1~I(Re^2)+St+Re+I(Re^2)*St+St:Re,data=data_train)
summary(model_1)
```

```
##
## Call:
## lm(formula = R_moment_1 ~ I(Re^2) + St + Re + I(Re^2) * St +
##     St:Re, data = data_train)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0274911 -0.0003305 -0.0000063  0.0001455  0.0298379
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.822e-01  7.757e-03  23.492  < 2e-16 ***
```

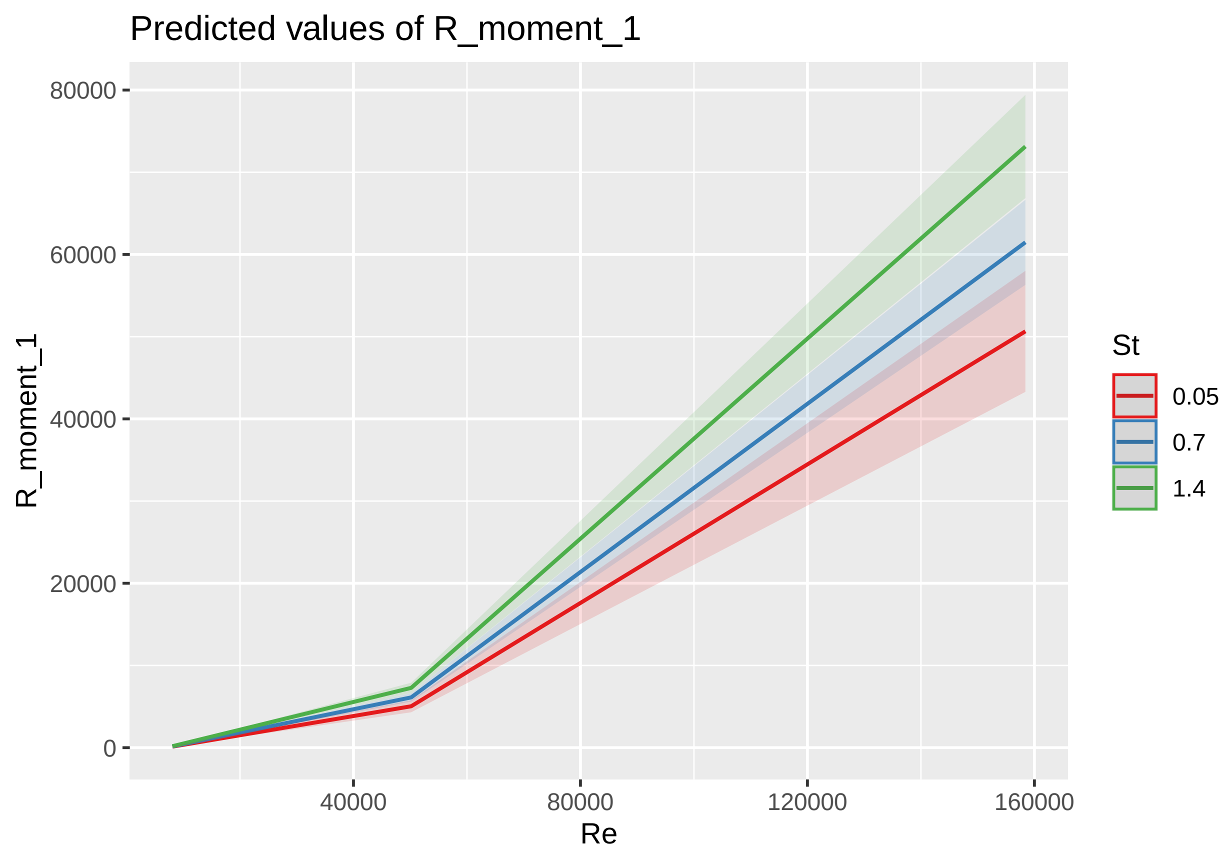
```
## I(Re^2)       1.993e-06  1.553e-07  12.838  < 2e-16 ***
## St            6.143e-02  6.487e-03   9.470 7.50e-15 ***
## Re           -1.250e-03  7.592e-05 -16.471  < 2e-16 ***
## I(Re^2):St    6.666e-07  1.364e-07   4.887 4.93e-06 ***
## St:Re        -4.195e-04  6.588e-05  -6.368 1.01e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01127 on 83 degrees of freedom
## Multiple R-squared:  0.9616, Adjusted R-squared:  0.9593
## F-statistic: 415.7 on 5 and 83 DF,  p-value: < 2.2e-16
```

As we can see a model with 5 variables is enough to predict the first raw moment accurately. Interestingly enough, similar to the linear models TFr doesn't appear to have a significant relationship between it and the first raw moment.

```
plot_model(model_1, type = "pred", terms = c("St", "Re[90,224,398]"))
```




```
plot_model(model_1, type = "pred", terms = c("Re","St[.05,.7,1.4]"))
```



This plot of the effect of Re on the first moment using the nonlinear model with interactions sheds light on the relationship between Re and the first moment. First, as we determined with the simple linear model, St has a positive relationship with the first moment. The impact of Re on the first moment is higher at each level of Re if St increases. However, this plot shows that as Re grows beyond roughly 50000, its marginal influence on the first moment increases. Thus, at very high values R seems to have a nonlinear relationship to the average expected amount of clustering.

```
model_2<-lm(R_moment_2~Re*TFr*St+I(TFr^2):I(Re^2)+I(TFr^2):St +I(TFr^2)+I(Re^2)-Re:TFr:St,data=data_trai
summary(model_2)
```

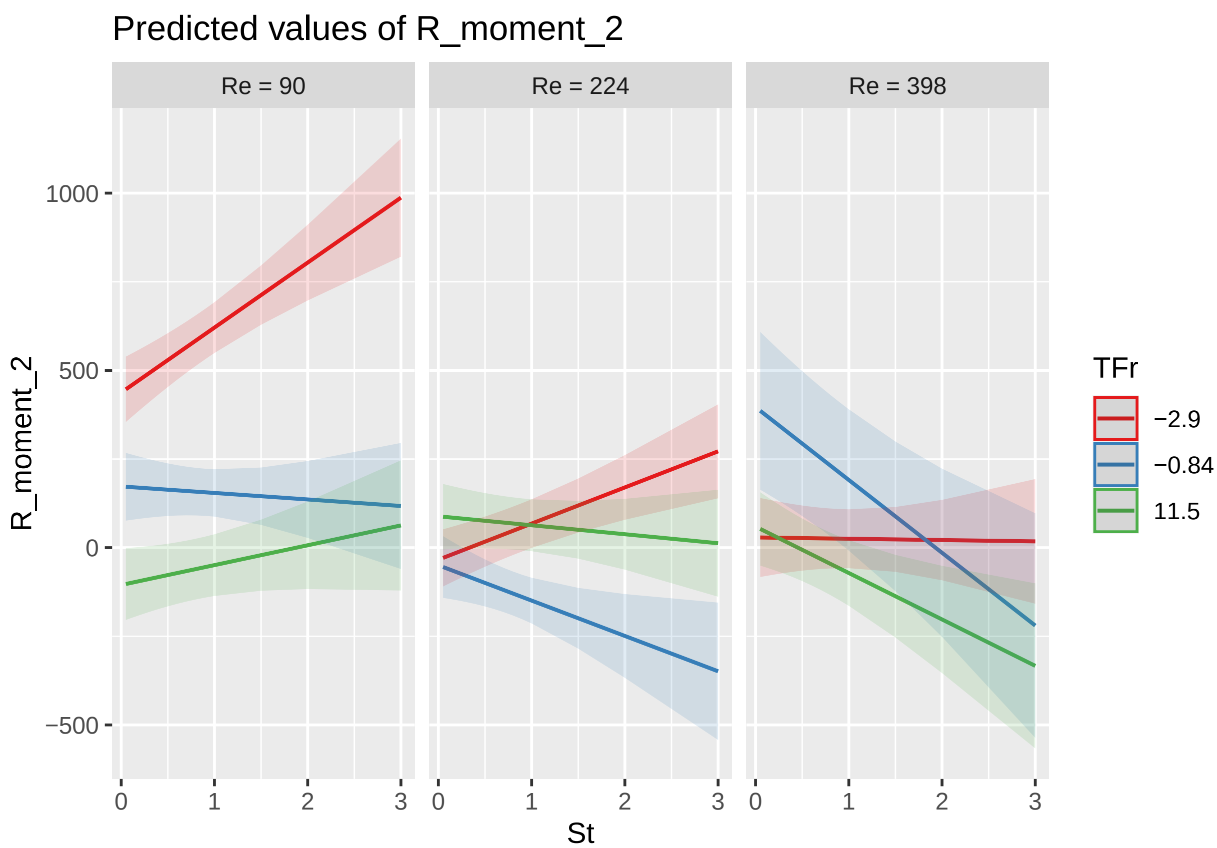
```
##
## Call:
## lm(formula = R_moment_2 ~ Re * TFr * St + I(TFr^2):I(Re^2) +
##     I(TFr^2):St + I(TFr^2) + I(Re^2) - Re:TFr:St, data = data_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -446.53  -67.92  -17.88   79.33  283.59
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.519e+02  9.369e+01   4.824 6.84e-06 ***
## Re               -5.346e+00  7.367e-01  -7.256 2.55e-10 ***
## TFr              -1.670e+02  2.530e+01  -6.600 4.49e-09 ***
## St               -2.838e+01  5.586e+01  -0.508 0.612807
## I(TFr^2)          8.527e+00  2.235e+00   3.815 0.000271 ***
```

```
## I(Re^2)           1.381e-02  2.023e-03   6.825 1.70e-09 ***
## Re:TFr            7.336e-01  1.024e-01   7.165 3.81e-10 ***
## Re:St            -6.071e-01  1.722e-01  -3.525 0.000713 ***
## TFr:St           -7.090e+01  1.746e+01  -4.060 0.000116 ***
## I(TFr^2):I(Re^2) -1.440e-04  2.371e-05  -6.074 4.27e-08 ***
## St:I(TFr^2)       7.216e+00  1.910e+00   3.778 0.000307 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 134.5 on 78 degrees of freedom
## Multiple R-squared:  0.766,  Adjusted R-squared:  0.736
## F-statistic: 25.53 on 10 and 78 DF,  p-value: < 2.2e-16
```

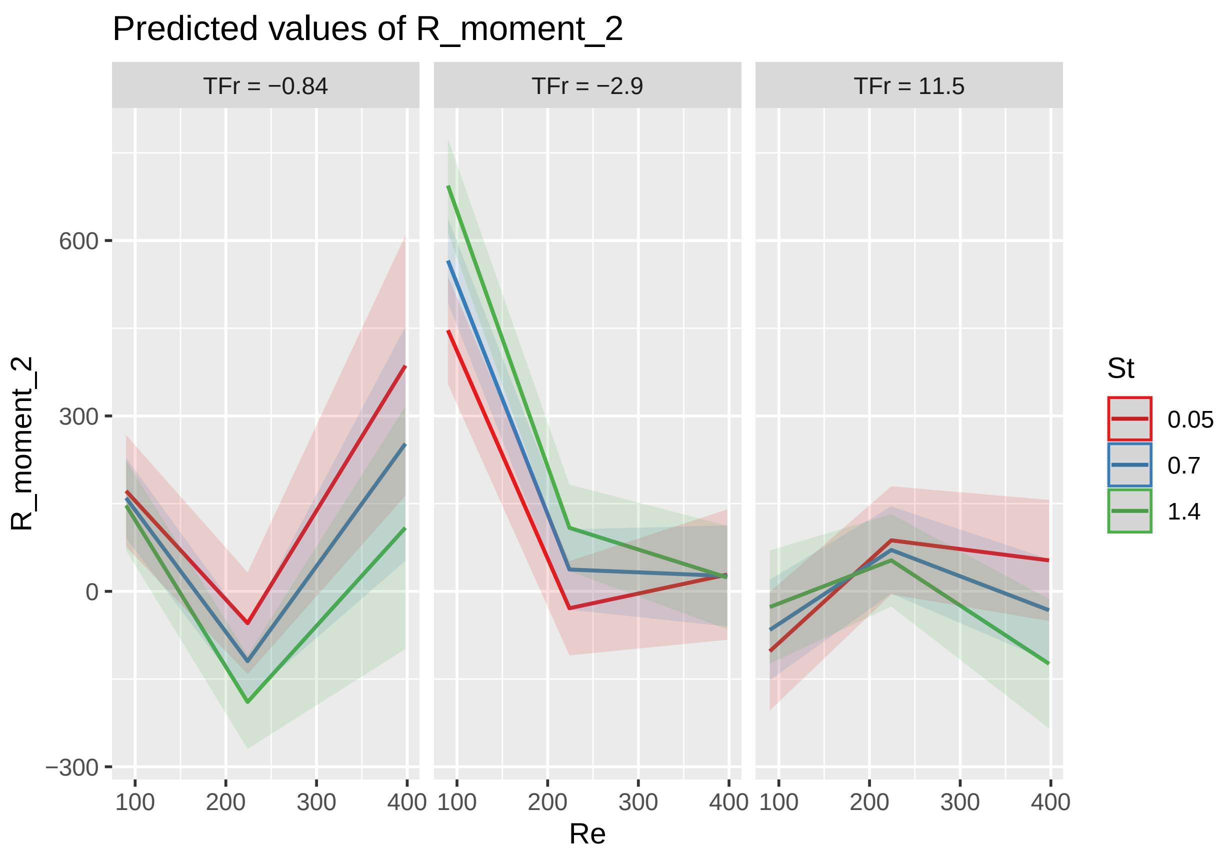
The final model for the second raw moment is a model with 10 variables that has up to two interaction levels.

```
plot_model(model_2, type = "pred", terms = c("TFr", "St[.05,.7,1.4]","Re[90,224,398]")) ##checking 500
```


```
plot_model(model_2, type = "pred", terms = c("St", "TFr[-.84,-2.9,11.5]", "Re[90,224,398]"))
```




```
plot_model(model_2, type = "pred", terms = c("Re", "St[.05,.7,1.4]", "TFr[-.84,-2.9,11.5]"))
```



Third plot series: From these plots of the second model, we can first notice a significant amount of overlap of error bounds in all three plots regardless of the level of St. Thus, when it comes to variance, the influence of the size of the particles is not particularly important. We can also see that the convergence of the fits tightens at TFr = 11.5 (Fr = inf). Because Fr = u/sqrt(gL), it makes sense that u (flow velocity) must be relatively large for Fr to be infinite (find any source). At high rates of flow may consistently break up the clusters and thus preventing the occasional examples of high levels of clustering. We also see that high Fr may limit the influence of Re. At near-zero Fr (TFr = -.84), high Re has a strong positive relationship with variance. Additionally, Re seems to exhibit some kind of thresholding behavior past Re = 224 where its relationship with variance reverses past this point. Re = 224 appears to be a special value worth studying more in the future.

Second plot series: These plots show us a relationship that later holds true in the second and third moments as well. First, we can see by the blue and green lines that regardless of the level of Re and St, higher levels of Fr decrease the variance of clustering. We saw this relationship in the previous set of plots. The most interesting line to observe is the red one that shows what happens to variance as St changes at near-zero Fr. As Re increases, the effect of St converges to essentially nothing. Combined with what we learn about the first moment (that at extremely high levels of Re, average clustering increases), perhaps this means that Re becomes the most important determinant of clustering as it reaches high levels. At very high Re, we may see consistent clustering regardless of the other variables. Nonetheless, if Re is low (90 or 224), St contributes to increases in variance. Thus, at low Re and Fr, not only does St increase average clustering (as we observed in the linear models), it does so with a great deal of variance. This probably reflects the fact that particles colliding does increase in frequency as they grow in size, but collisions are very unpredictable and random events; sometimes they lead to clustering and other times they do not.

```
coef(step.model.two$finalModel, 9)
```

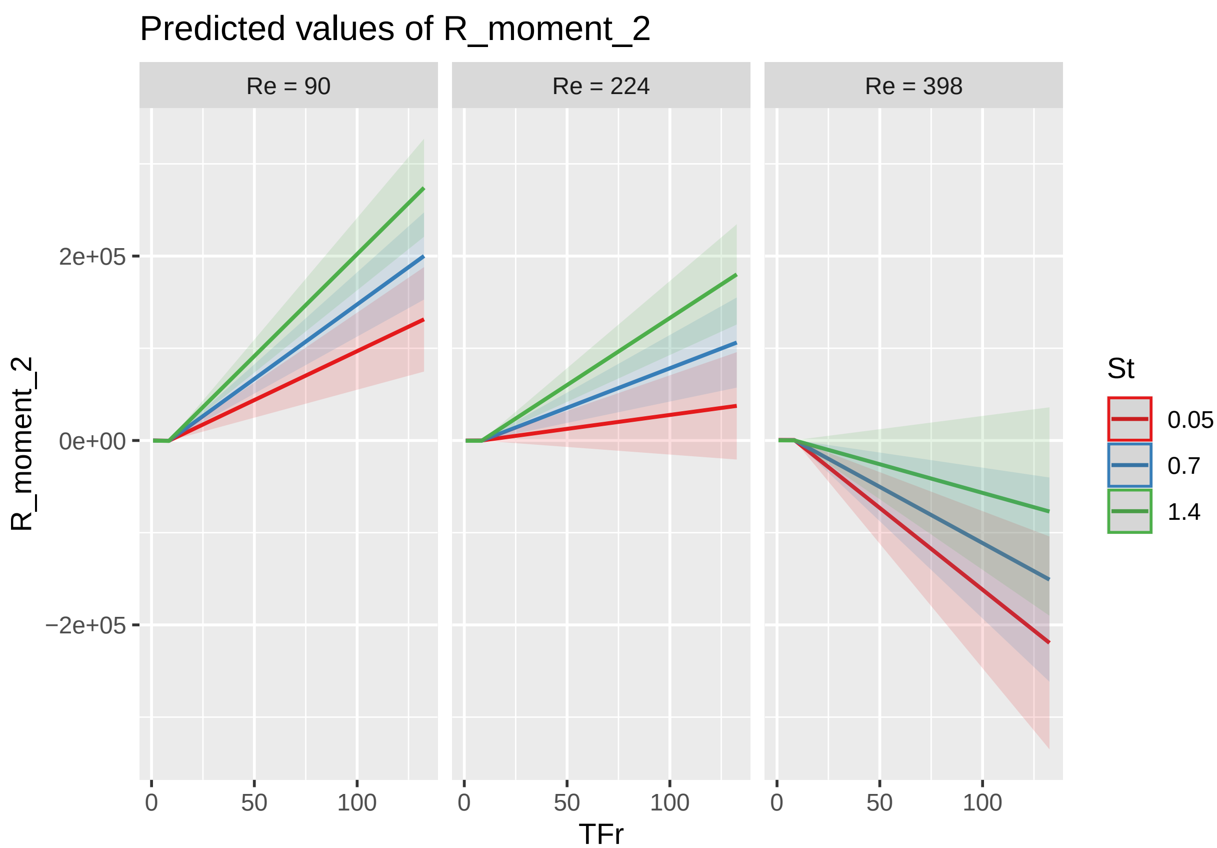
```
##       (Intercept)          I(TFr^2)          I(Re^2)                 Re
```

```
##     4.239702e+02     9.146369e+00     1.377607e-02    -5.293138e+00
##              TFr I(TFr^2):I(Re^2)      I(TFr^2):St            St:Re
##    -1.724373e+02    -1.444094e-04     6.509192e+00    -6.362037e-01
##           St:TFr           Re:TFr
##    -6.471653e+01     7.345131e-01
```

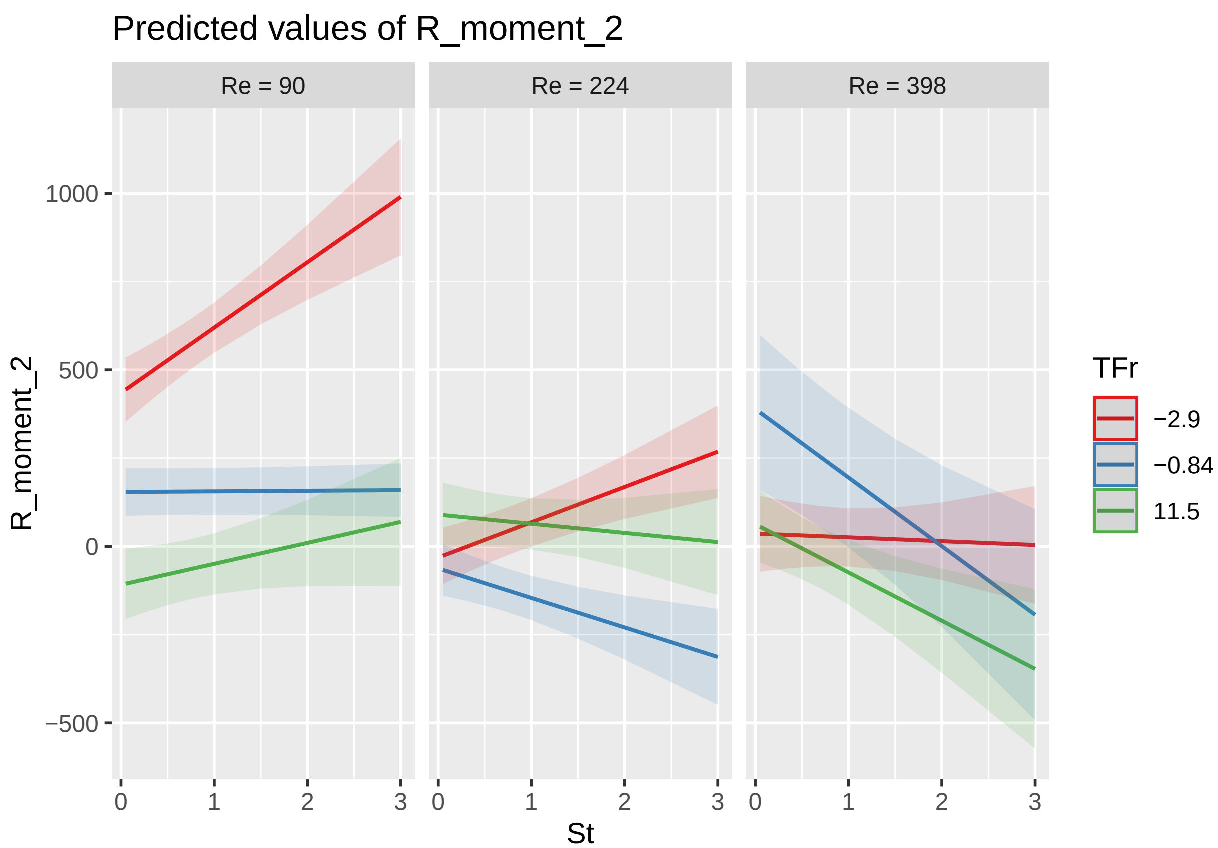
```
model_2_alt <- lm(R_moment_2~I(TFr^2)*I(Re^2) + I(TFr^2):St + St:Re + St:TFr + Re*TFr, data=data_train)
summary(model_2_alt)
```

```
##
## Call:
## lm(formula = R_moment_2 ~ I(TFr^2) * I(Re^2) + I(TFr^2):St +
##     St:Re + St:TFr + Re * TFr, data = data_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -443.83  -70.52  -15.86   87.34  283.59
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.240e+02  7.544e+01   5.620 2.77e-07 ***
## I(TFr^2)          9.146e+00  1.865e+00   4.904 4.93e-06 ***
## I(Re^2)           1.378e-02  2.013e-03   6.845 1.48e-09 ***
## Re               -5.293e+00  7.259e-01  -7.292 2.06e-10 ***
## TFr              -1.724e+02  2.284e+01  -7.550 6.52e-11 ***
## I(TFr^2):I(Re^2) -1.444e-04  2.358e-05  -6.123 3.34e-08 ***
## I(TFr^2):St       6.509e+00  1.302e+00   4.998 3.40e-06 ***
## St:Re            -6.362e-01  1.617e-01  -3.935 0.000178 ***
## St:TFr           -6.472e+01  1.247e+01  -5.191 1.59e-06 ***
## Re:TFr            7.345e-01  1.019e-01   7.209 2.97e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 133.9 on 79 degrees of freedom
## Multiple R-squared:  0.7652, Adjusted R-squared:  0.7385
## F-statistic: 28.61 on 9 and 79 DF,  p-value: < 2.2e-16
```

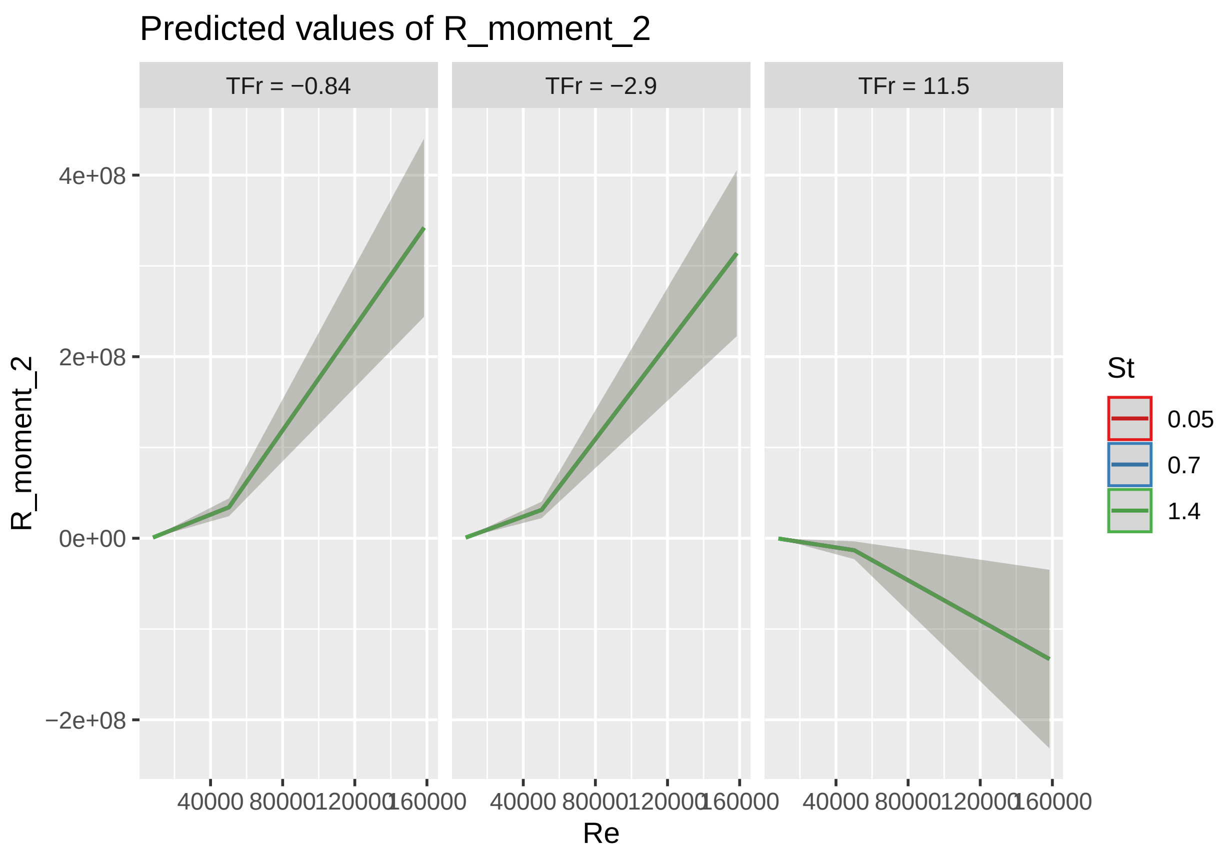
```
plot_model(model_2_alt, type = "pred", terms = c("TFr", "St[.05,.7,1.4]","Re[90,224,398]"))
```




```
plot_model(model_2_alt, type = "pred", terms = c("St", "TFr[-.84,-2.9,11.5]", "Re[90,224,398]"))
```




```
plot_model(model_2_alt, type = "pred", terms = c("Re", "St[.05,.7,1.4]", "TFr[-.84,-2.9,11.5]"))
```

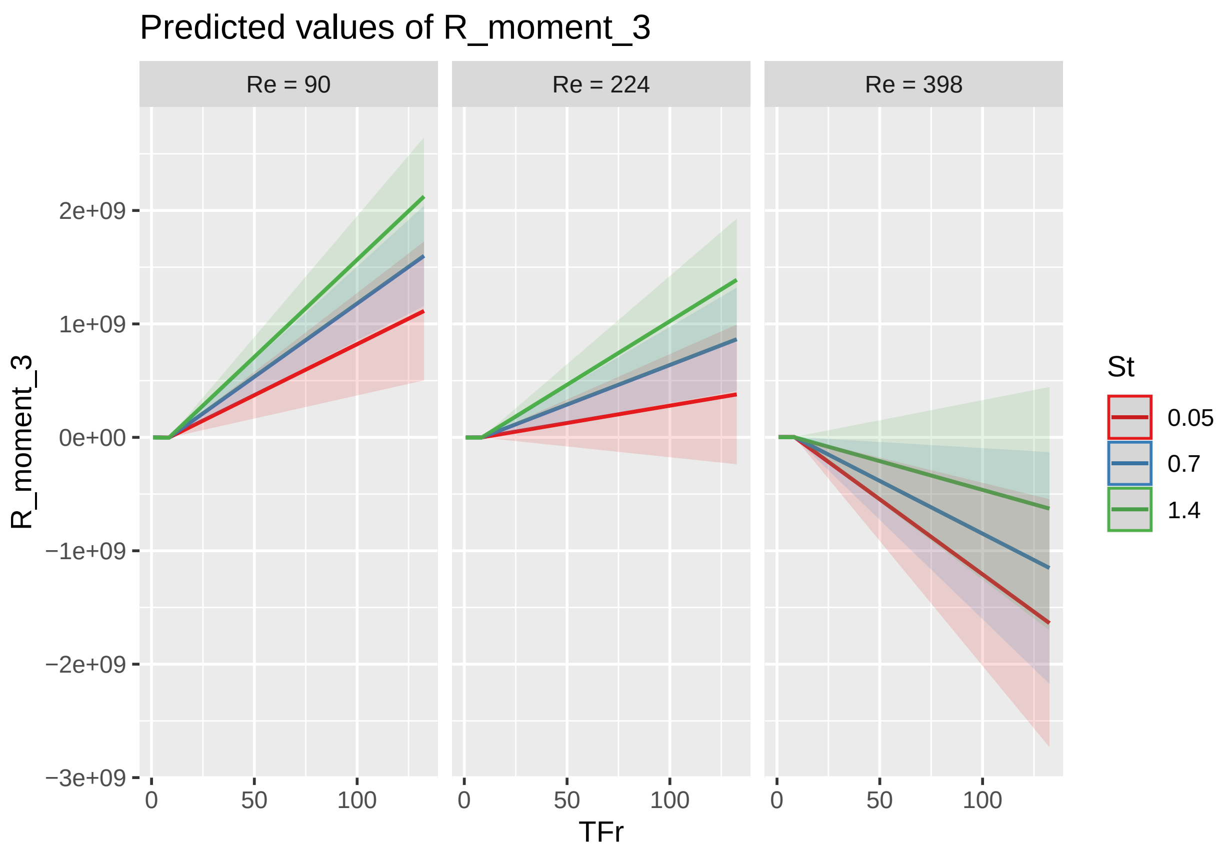



```
model_3<-lm(R_moment_3~I(TFr^2)*I(Re^2)+St*TFr+Re*TFr+I(TFr^2):St,data=data_train)
summary(model_3)
```

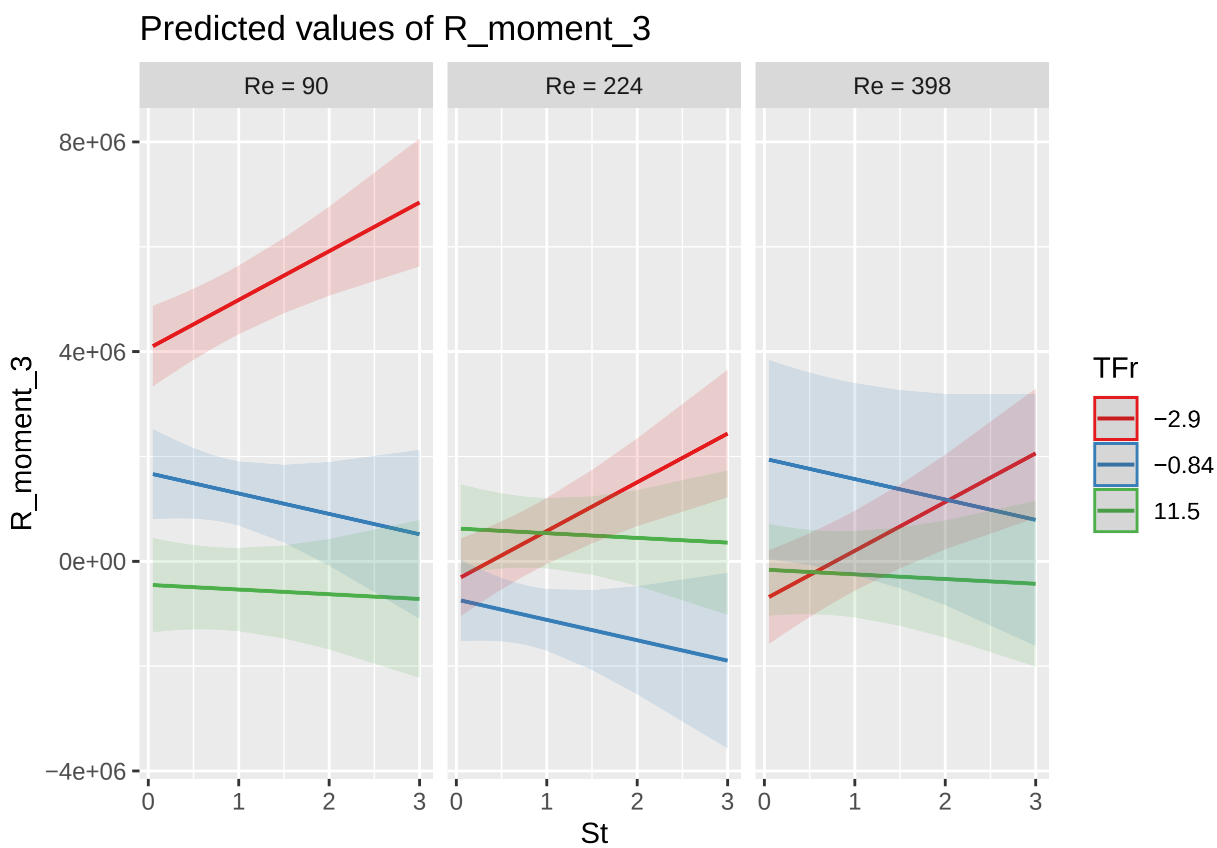
```
##
## Call:
## lm(formula = R_moment_3 ~ I(TFr^2) * I(Re^2) + St * TFr + Re *
##     TFr + I(TFr^2):St, data = data_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4107519  -580683   194083   640231  3090996
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.236e+06  8.449e+05   5.013 3.21e-06 ***
## I(TFr^2)          7.720e+04  2.021e+04   3.820 0.000265 ***
## I(Re^2)           1.093e+02  1.856e+01   5.890 8.99e-08 ***
## St               -8.149e+05  4.841e+05  -1.683 0.096237 .
## TFr              -1.429e+06  2.300e+05  -6.214 2.28e-08 ***
## Re               -4.711e+04  6.672e+03  -7.062 5.69e-10 ***
## I(TFr^2):I(Re^2) -1.134e+00  2.175e-01  -5.216 1.43e-06 ***
## St:TFr           -4.678e+05  1.523e+05  -3.072 0.002917 **
## TFr:Re            5.905e+03  9.404e+02   6.280 1.72e-08 ***
## I(TFr^2):St       4.616e+04  1.667e+04   2.769 0.007001 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1236000 on 79 degrees of freedom
## Multiple R-squared:  0.7122, Adjusted R-squared:  0.6794
## F-statistic: 21.72 on 9 and 79 DF,  p-value: < 2.2e-16
```

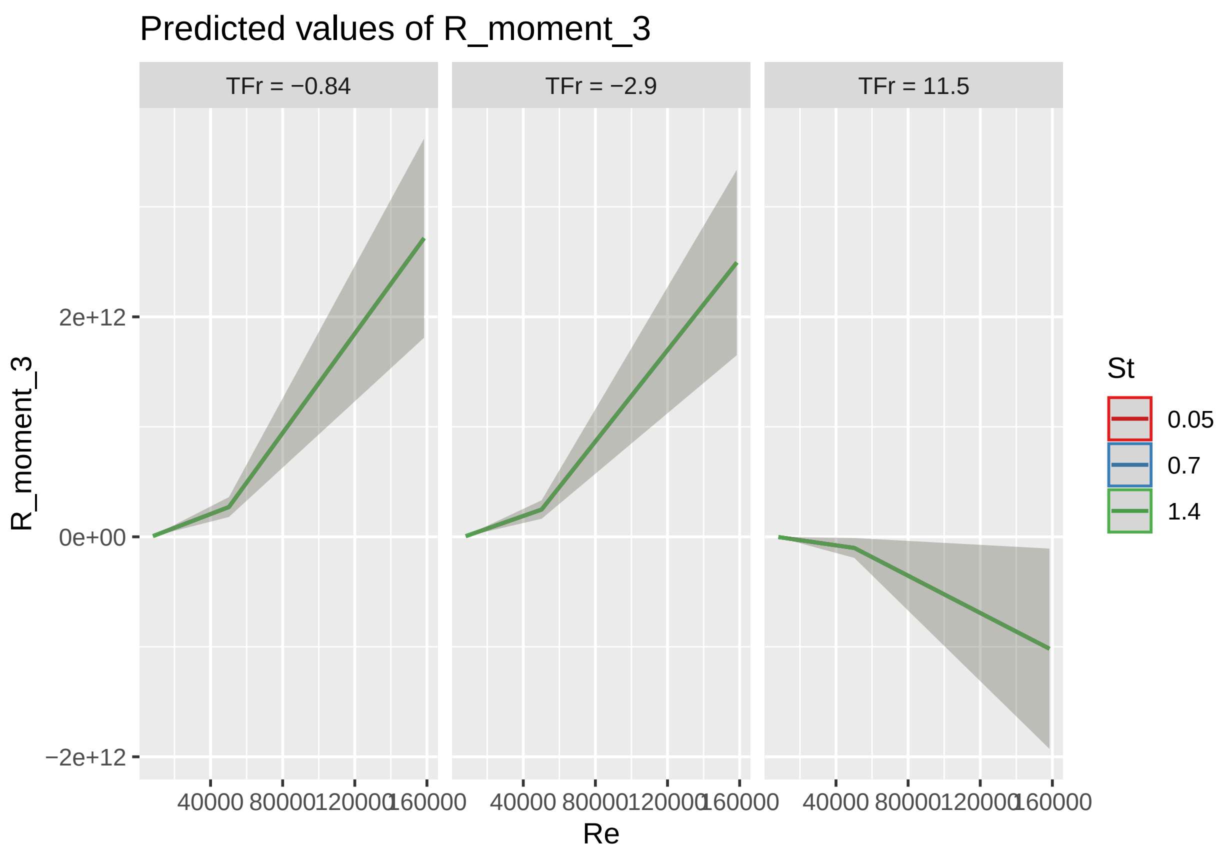
```
plot_model(model_3, type = "pred", terms = c("TFr", "St[.05,.7,1.4]","Re[90,224,398]"))
```




```
plot_model(model_3, type = "pred", terms = c("St", "TFr[-.84,-2.9,11.5]", "Re[90,224,398]"))
```




```
plot_model(model_3, type = "pred", terms = c("Re", "St[.05,.7,1.4]", "TFr[-.84,-2.9,11.5]"))
```



Second plot: Once again, we see essentially the same relationship between St and the third moment as we do between St and the second moment. Thus, higher Fr and higher Re decreases skewness. However, the slope of the red lines is always positive. Thus, in this case, St appears to push the distribution to the left (positive skewness means left lean and negative skewness means right lean) as variance decreases due to Fr and Re. Thus, we can see that in a limited way, particle size decreases clustering within certain fixed conditions.

```
coef(step.model.three$finalModel, 10)
```

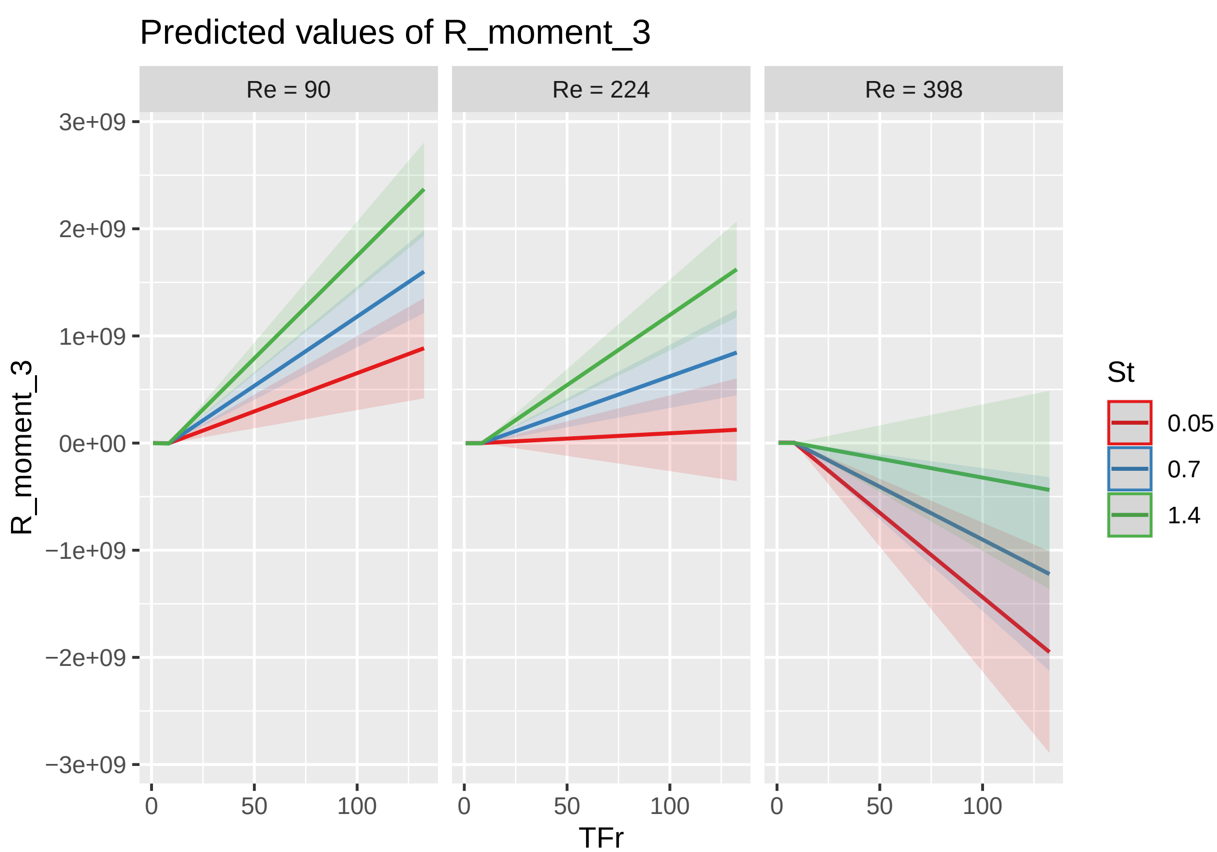
```
##       (Intercept)          I(TFr^2)          I(Re^2)                Re
##      3.463638e+06      6.185401e+04     1.134114e+02     -4.224438e+04
##               TFr I(TFr^2):I(Re^2)      I(TFr^2):St             St:Re
##     -1.186503e+06     -1.160563e+00     6.821466e+04     -6.800008e+03
##            St:TFr            Re:TFr         St:Re:TFr
##     -7.873823e+05      5.472355e+03     5.661473e+02
```

```
model_3_alt <- lm(R_moment_3~I(TFr^2)+I(Re^2)+Re+TFr+I(TFr^2):I(Re^2)+I(TFr^2):St+St:Re+St:TFr+Re:TFr+S
summary(model_3_alt)
```

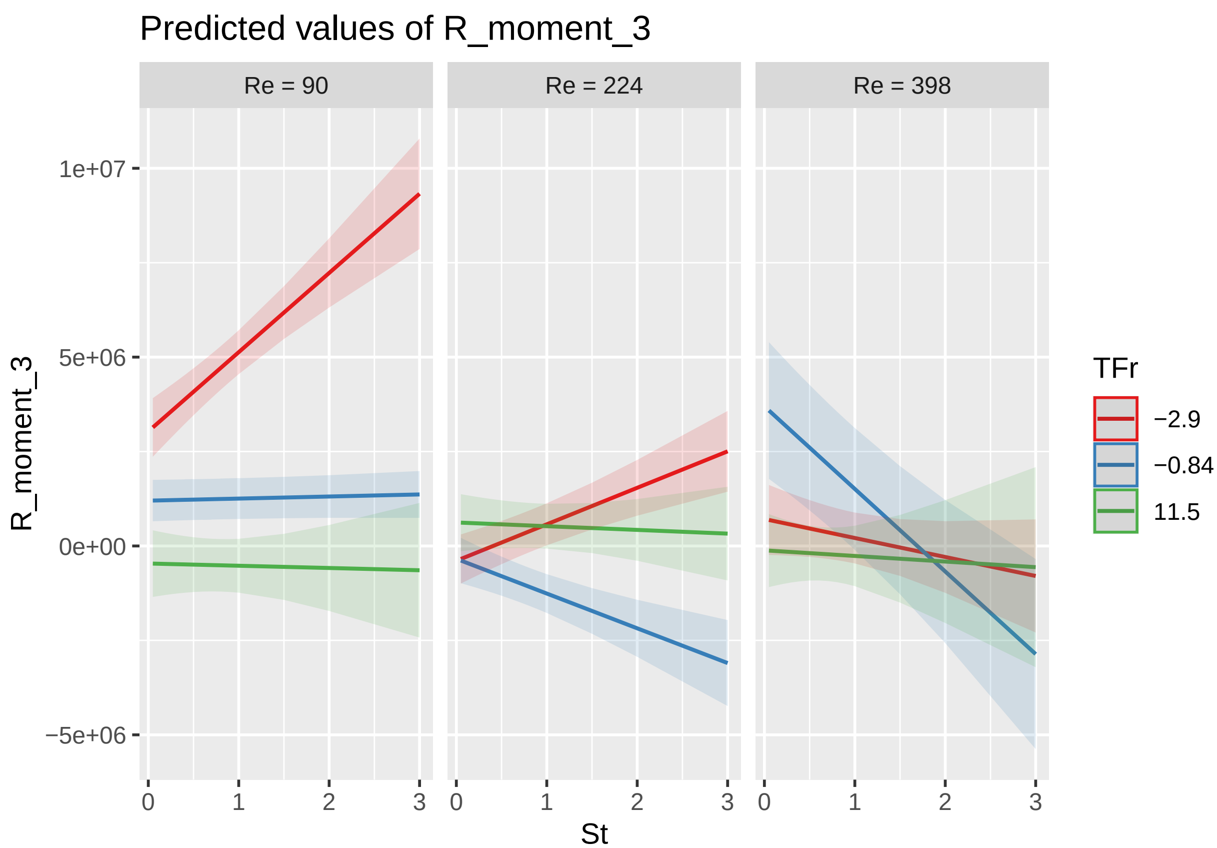
```
##
## Call:
## lm(formula = R_moment_3 ~ I(TFr^2) + I(Re^2) + Re + TFr + I(TFr^2):I(Re^2) +
##     I(TFr^2):St + St:Re + St:TFr + Re:TFr + St:Re:TFr, data = data_train)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -3142111  -563374   145198   500936  2361633
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.464e+06  6.151e+05   5.631 2.71e-07 ***
## I(TFr^2)          6.185e+04  1.542e+04   4.012 0.000137 ***
## I(Re^2)           1.134e+02  1.641e+01   6.911 1.16e-09 ***
## Re               -4.224e+04  5.919e+03  -7.137 4.32e-10 ***
## TFr              -1.187e+06  1.957e+05  -6.063 4.47e-08 ***
## I(TFr^2):I(Re^2) -1.161e+00  1.924e-01  -6.031 5.12e-08 ***
## I(TFr^2):St       6.821e+04  1.098e+04   6.213 2.37e-08 ***
## Re:St            -6.800e+03  1.363e+03  -4.988 3.61e-06 ***
## TFr:St           -7.874e+05  1.227e+05  -6.420 9.79e-09 ***
## Re:TFr            5.472e+03  8.544e+02   6.405 1.04e-08 ***
## Re:TFr:St         5.661e+02  2.121e+02   2.669 0.009258 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1091000 on 78 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:   0.75
## F-statistic: 27.41 on 10 and 78 DF,  p-value: < 2.2e-16
```

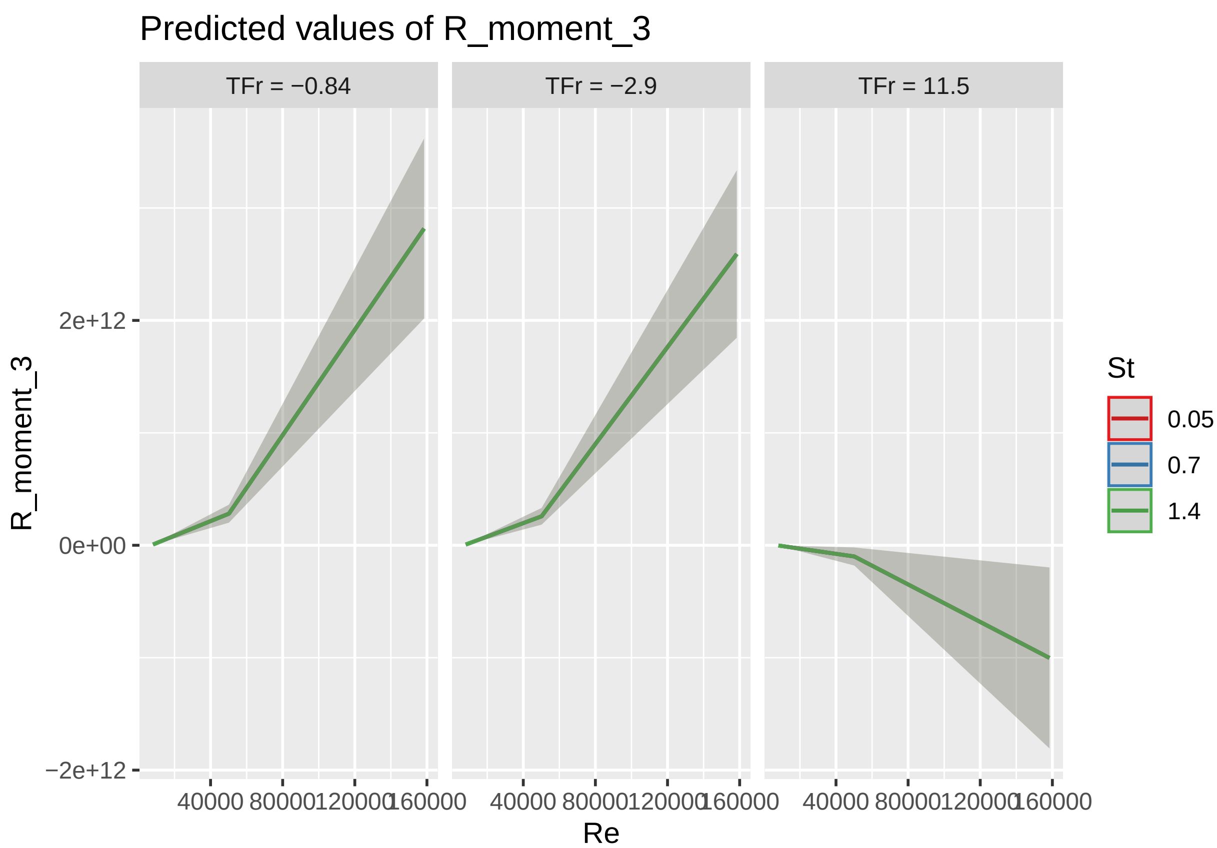
```
plot_model(model_3_alt, type = "pred", terms = c("TFr", "St[.05,.7,1.4]","Re[90,224,398]"))
```



```r
plot_model(model_3_alt, type = "pred", terms = c("St", "TFr[-.84,-2.9,11.5]", "Re[90,224,398]"))
```




```r
plot_model(model_3_alt, type = "pred", terms = c("Re", "St[.05,.7,1.4]", "TFr[-.84,-2.9,11.5]"))
```



Our Final model for Moment 4 will be the full model with 13 variables (ST is in the model due to the hierarchy principle) . This model has all interaction variables up to 3, since they appear to be significant (P<0.05)

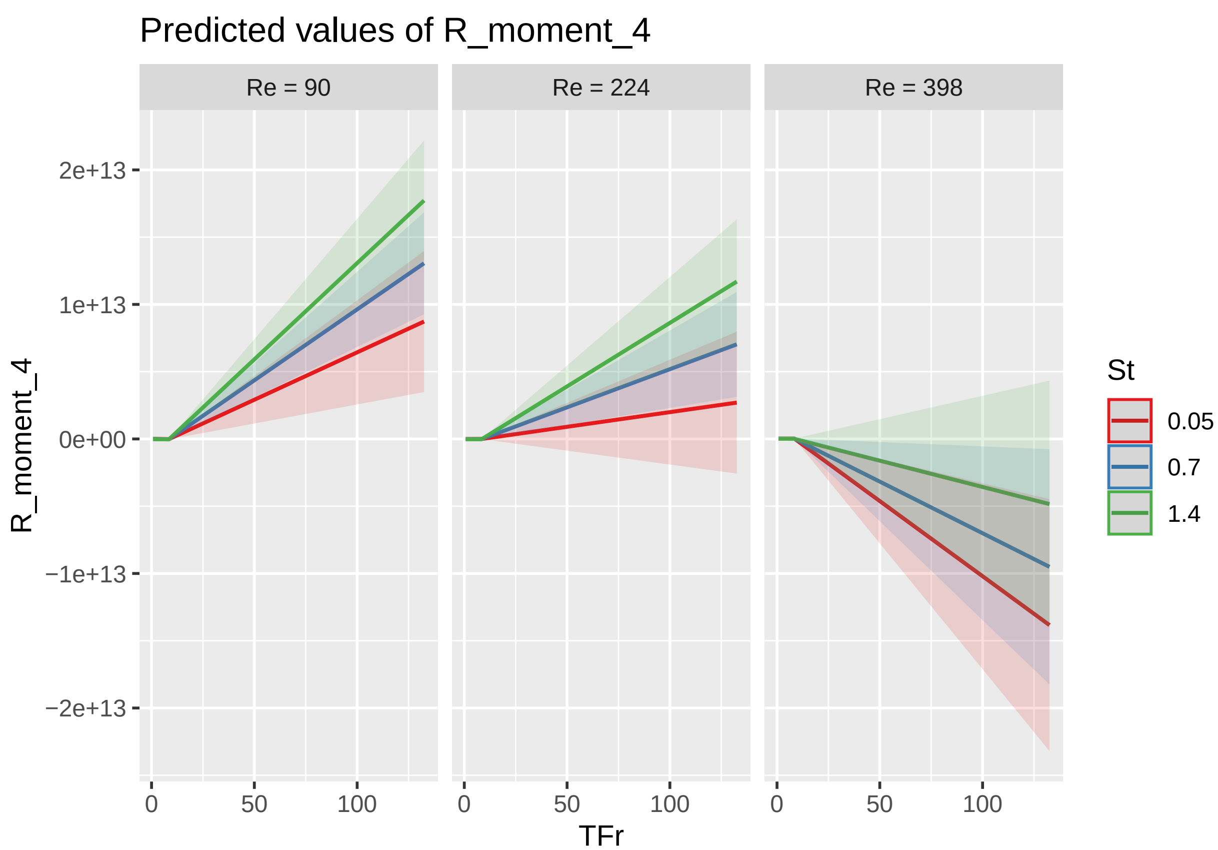
```
model_4<-lm(R_moment_4~I(TFr^2)*I(Re^2)+St*TFr+Re*TFr+I(TFr^2):St,data=data_train)
summary(model_4)
```

```
##
## Call:
## lm(formula = R_moment_4 ~ I(TFr^2) * I(Re^2) + St * TFr + Re *
##     TFr + I(TFr^2):St, data = data_train)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.298e+10 -4.906e+09  1.731e+09  5.650e+09  2.784e+10
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.504e+10  7.233e+09   4.845 6.20e-06 ***
## I(TFr^2)          6.062e+08  1.730e+08   3.503 0.000759 ***
## I(Re^2)           8.955e+05  1.589e+05   5.634 2.60e-07 ***
## St               -7.011e+09  4.144e+09  -1.692 0.094617 .
## TFr              -1.144e+10  1.969e+09  -5.811 1.25e-07 ***
## Re               -3.864e+08  5.711e+07  -6.765 2.10e-09 ***
## I(TFr^2):I(Re^2) -9.301e+03  1.862e+03  -4.996 3.43e-06 ***
## St:TFr           -4.185e+09  1.304e+09  -3.211 0.001917 **
```

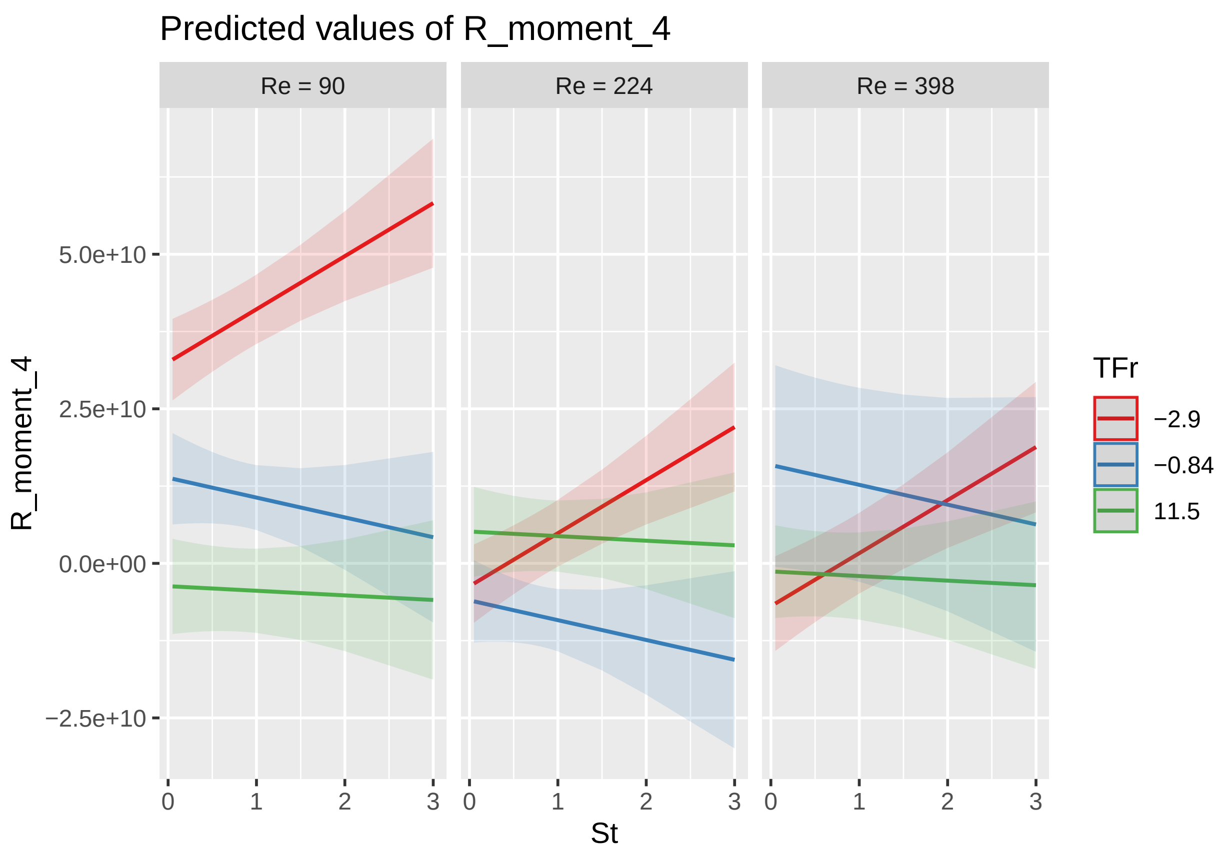
```
## TFr:Re             4.847e+07  8.050e+06   6.021 5.17e-08 ***
## I(TFr^2):St        4.113e+08  1.427e+08   2.882 0.005081 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.058e+10 on 79 degrees of freedom
## Multiple R-squared:  0.6992, Adjusted R-squared:  0.665
## F-statistic: 20.41 on 9 and 79 DF,  p-value: < 2.2e-16
```

Our Final model for Moment 4 will be the full model with 13 variables (ST is in the model due to the hierarchy principle) . This model has all interaction variables up to 3, since they appear to be significant (P<0.05).

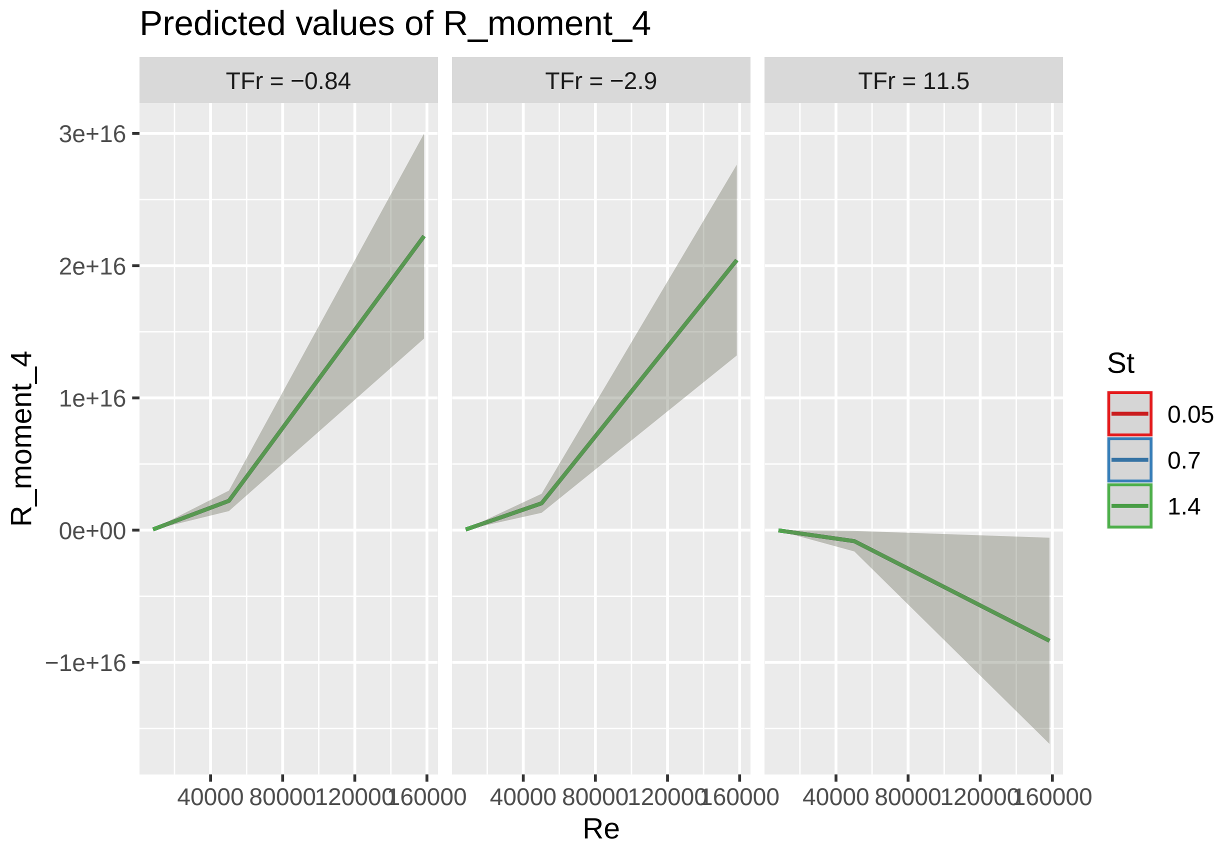
```
plot_model(model_4, type = "pred", terms = c("TFr", "St[.05,.7,1.4]","Re[90,224,398]"))
```




```
plot_model(model_4, type = "pred", terms = c("St", "TFr[-.84,-2.9,11.5]", "Re[90,224,398]"))
```




```
plot_model(model_4, type = "pred", terms = c("Re", "St[.05,.7,1.4]", "TFr[-.84,-2.9,11.5]"))
```



Second plot series: The relationship between the predictors and kurtosis essentially seems identical to their relationship with skewness. Thus, Re = 398 plot is interesting because in this plot for the second moment, we saw the effects of all variables converge to zero. Thus, even as variance is not very effected, the size of the particles still makes the tails of the probability distribution of clustering heavier. Similar to our analysis of St's relationship to the third moment, it seems to increase the spread of the distribution within bounds set by the other parameters. Thus, as the other parameters make tail weight smaller overall, higher St makes them as small as possible within those bounds. I believe that this makes sense because Fr and Re seem to be more related to the environment than St, which has to do with the particles themselves. Given certain environmental parameters, larger particle size will always increase average clustering but increase left-leaning skewness and tail weight within the limited variance determined by Re and Fr.

Discussion & Conclusion

Conclusion: In conclusion, we have learned that St and the first moment have a mostly positive and linear relationship, although predictive accuracy can increase if we use a more complex model that includes interactions. Through the model with interactions, we saw that extremely high values of Re will increase average clustering. The other moments are best fitted with models that involve complex interactions and exponential terms due to their inherent nonlinearity. In general, high levels of Fr and Re decrease variance, skewness, and kurtosis. These parameters seem to make the results of the simulation more regular and consistent. On the other hand, St pushes for more positive skewness and kurtosis. I believe that this probably has to do with the inherently chaotic and unpredictable nature of collisions between particles. Lastly, with regards to variance, we saw Re = 224 exhibits some kind of thresholding behavior because its effect on variance switches at that point. In addition to this mystery, the specific relationship between Re and Fr is also worth further consideration. Intuitively, as Fr increases I would expect clustering behavior to become more regular because high flow probably cuts short any particularly unusual behavior of particles. However, Re when combined with Fr also seems to concentrate the probability of clustering around the mean. Theoretically, I would expect Re to both increase mean clustering and lead to less predictable behavior. Perhaps our results mean that high levels of Re in fact leads to more clustering, albeit in very consistent ways.

References

https://www.britannica.com/science/Reynolds-number

Smaller Models

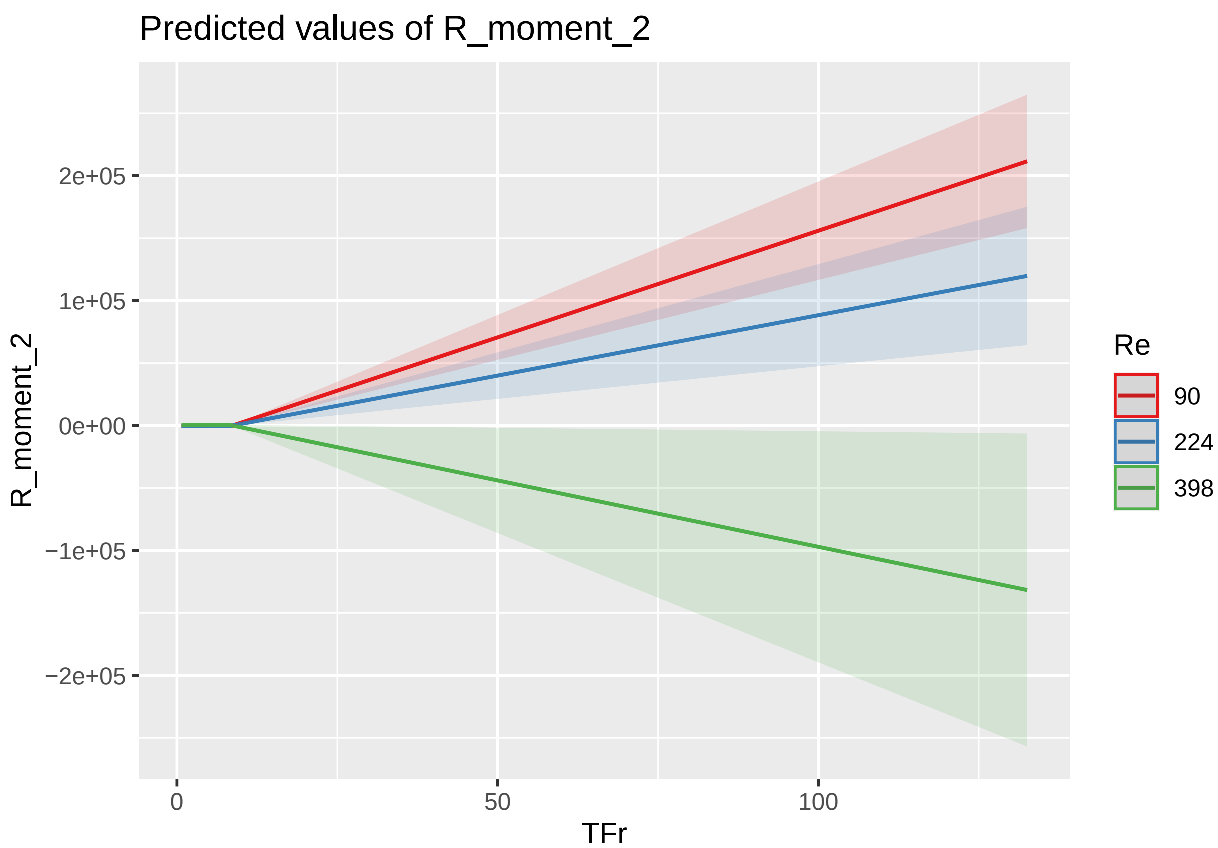
```
coef(step.model.two$finalModel, 6)
```

```
##      (Intercept)         I(TFr^2)          I(Re^2)               Re
##     4.323929e+02     1.438803e+01     1.364767e-02    -5.808202e+00
##              TFr I(TFr^2):I(Re^2)           Re:TFr
##    -2.254036e+02    -1.413402e-04     7.297198e-01
```

```
model_2_small <- lm(R_moment_2~I(TFr^2)*I(Re^2)+Re*TFr,data=data_train)
summary(model_2_small)
```

```
##
## Call:
## lm(formula = R_moment_2 ~ I(TFr^2) * I(Re^2) + Re * TFr, data = data_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -594.99  -69.98  -26.36   64.08  448.77
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.324e+02  8.570e+01   5.045 2.67e-06 ***
## I(TFr^2)          1.439e+01  1.747e+00   8.235 2.39e-12 ***
## I(Re^2)           1.365e-02  2.280e-03   5.986 5.45e-08 ***
## Re               -5.808e+00  8.174e-01  -7.106 3.99e-10 ***
## TFr              -2.254e+02  2.319e+01  -9.722 2.65e-15 ***
## I(TFr^2):I(Re^2) -1.413e-04  2.663e-05  -5.307 9.32e-07 ***
## Re:TFr            7.297e-01  1.153e-01   6.327 1.25e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 152.3 on 82 degrees of freedom
## Multiple R-squared:  0.6846, Adjusted R-squared:  0.6615
## F-statistic: 29.66 on 6 and 82 DF,  p-value: < 2.2e-16
```

```
plot_model(model_2_small, type = "pred", terms = c("TFr", "Re[90,224,398]"))
```




```
coef(step.model.three$finalModel, 6)
```

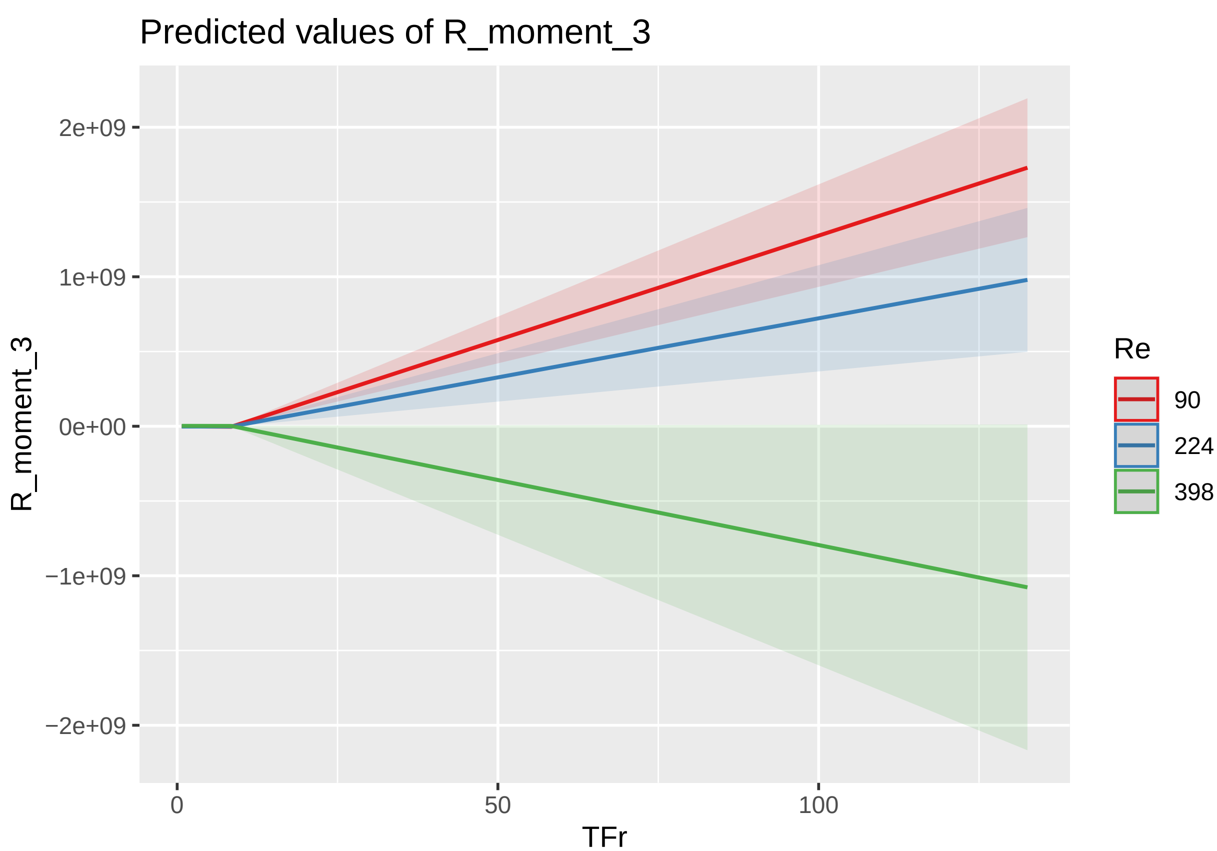
```
##      (Intercept)          I(TFr^2)          I(Re^2)                Re
##     3.522528e+06      1.176447e+05     1.115015e+02     -4.741731e+04
##              TFr I(TFr^2):I(Re^2)           Re:TFr
##    -1.842971e+06     -1.156087e+00     5.967790e+03
```

```
model_3_small <- lm(R_moment_3~I(TFr^2)*I(Re^2)+Re*TFr,data=data_train)
summary(model_3_small)
```

```
##
## Call:
## lm(formula = R_moment_3 ~ I(TFr^2) * I(Re^2) + Re * TFr, data = data_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4861786  -572778  -215754   523977  4278191
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.523e+06  7.453e+05   4.726 9.37e-06 ***
## I(TFr^2)          1.176e+05  1.519e+04   7.742 2.26e-11 ***
## I(Re^2)           1.115e+02  1.983e+01   5.623 2.52e-07 ***
## Re               -4.742e+04  7.109e+03  -6.670 2.77e-09 ***
```

```
## TFr                 -1.843e+06  2.016e+05  -9.140 3.79e-14 ***
## I(TFr^2):I(Re^2) -1.156e+00  2.316e-01  -4.991 3.32e-06 ***
## Re:TFr               5.968e+03  1.003e+03   5.950 6.35e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1325000 on 82 degrees of freedom
## Multiple R-squared:  0.6568, Adjusted R-squared:  0.6317
## F-statistic: 26.16 on 6 and 82 DF,  p-value: < 2.2e-16
```

```
plot_model(model_3_small, type = "pred", terms = c("TFr", "Re[90,224,398]"))
```




```
coef(step.model.four$finalModel, 6)
```

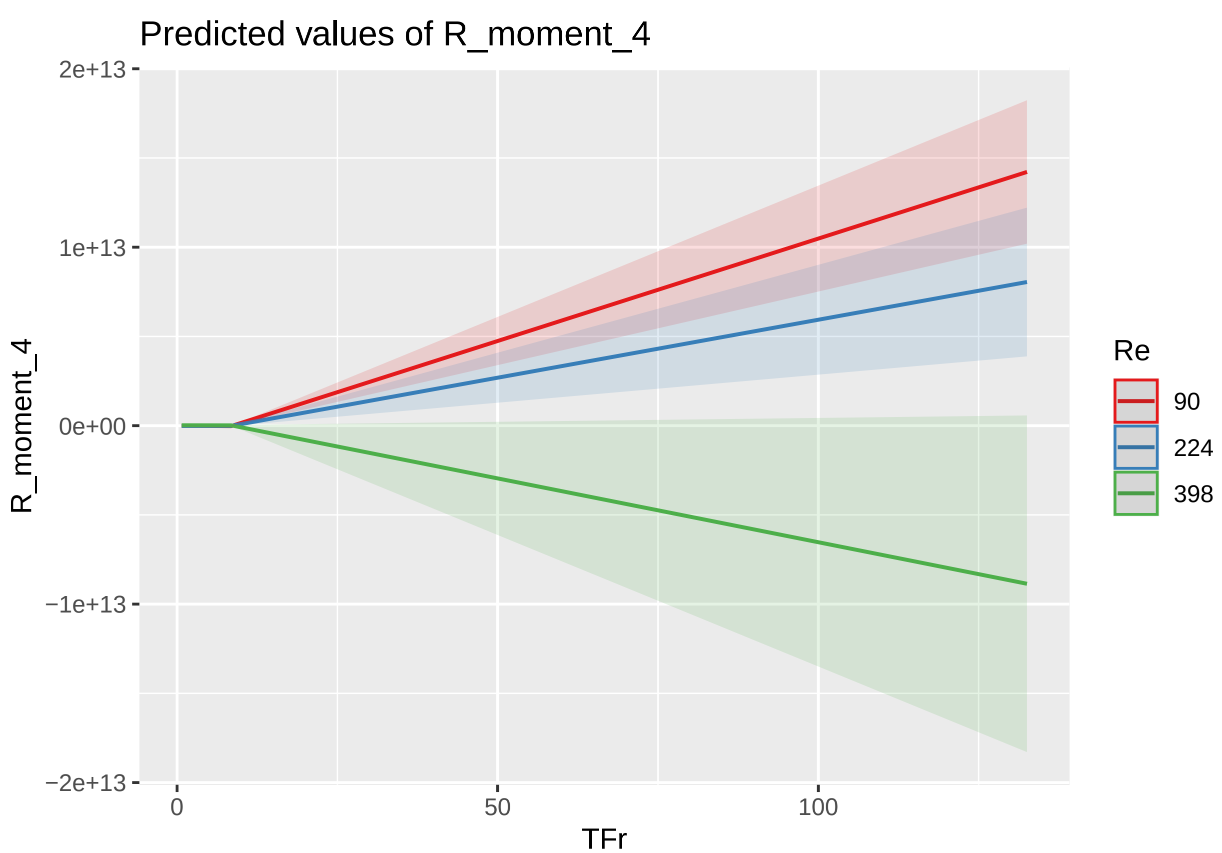
```
##      (Intercept)         I(TFr^2)          I(Re^2)               Re
##     2.896209e+10     9.672622e+08     9.167672e+05    -3.898643e+08
##              TFr I(TFr^2):I(Re^2)           Re:TFr
##    -1.515276e+10    -9.505362e+03     4.906712e+07
```

```
model_4_small <- lm(R_moment_4~I(TFr^2)*I(Re^2)+Re*TFr,data=data_train)
summary(model_4_small)
```

```
##
## Call:
```

```
## lm(formula = R_moment_4 ~ I(TFr^2) * I(Re^2) + Re * TFr, data = data_train)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.997e+10 -4.709e+09 -1.774e+09  4.308e+09  4.003e+10
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.896e+10  6.454e+09   4.487 2.33e-05 ***
## I(TFr^2)           9.673e+08  1.316e+08   7.351 1.33e-10 ***
## I(Re^2)            9.168e+05  1.717e+05   5.339 8.16e-07 ***
## Re                -3.899e+08  6.156e+07  -6.333 1.22e-08 ***
## TFr               -1.515e+10  1.746e+09  -8.678 3.14e-13 ***
## I(TFr^2):I(Re^2) -9.505e+03  2.006e+03  -4.739 8.92e-06 ***
## Re:TFr             4.907e+07  8.686e+06   5.649 2.26e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.147e+10 on 82 degrees of freedom
## Multiple R-squared:  0.6331, Adjusted R-squared:  0.6063
## F-statistic: 23.58 on 6 and 82 DF,  p-value: 5.21e-16
```

```
plot_model(model_4_small, type = "pred", terms = c("TFr", "Re[90,224,398]"))
```



‘‘