STA 325 Case Study

Load libraries and data

```
## # A tibble: 89 x 7
##       St    Re      Fr R_moment_1 R_moment_2 R_moment_3 R_moment_4
##    <dbl> <dbl>   <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
##  1  0.1    224   0.052   0.00216     0.130      14.4       1586.
##  2  3      224   0.052   0.00379     0.470      69.9      10404
##  3  0.7    224 Inf       0.00291     0.0435      0.822       15.6
##  4  0.05    90 Inf       0.0635      0.0907      0.467        3.27
##  5  0.7    398 Inf       0.000369    0.00622     0.126        2.57
##  6  2       90   0.3     0.148       2.01       36.2        672.
##  7  0.2     90 Inf       0.0813      0.324       3.04        33.0
##  8  3      224 Inf       0.00575     0.120       2.75        63.2
##  9  0.9    224 Inf       0.00302     0.0452      0.845       15.8
## 10  0.6    398   0.052   0.000314    0.00447     0.0821       1.51
## # ... with 79 more rows
```

```
##                      St          Re  Fr R_moment_1 R_moment_2 R_moment_3
## St          1.00000000 -0.03169871 NaN  0.2147681  0.1479257  0.1647465
## Re         -0.03169871  1.00000000 NaN -0.7747206 -0.3932344 -0.3844289
## Fr                 NaN         NaN   1        NaN        NaN        NaN
## R_moment_1  0.21476813 -0.77472058 NaN  1.0000000  0.6298829  0.6217326
## R_moment_2  0.14792571 -0.39323445 NaN  0.6298829  1.0000000  0.9984335
## R_moment_3  0.16474648 -0.38442895 NaN  0.6217326  0.9984335  1.0000000
## R_moment_4  0.18004537 -0.37741773 NaN  0.6150484  0.9946671  0.9988414
##            R_moment_4
## St          0.1800454
## Re         -0.3774177
## Fr                NaN
## R_moment_1  0.6150484
## R_moment_2  0.9946671
## R_moment_3  0.9988414
## R_moment_4  1.0000000
```

```
## # A tibble: 23 x 3
##       St    Re      Fr
##    <dbl> <dbl>   <dbl>
##  1  0.05   398   0.052
##  2  0.2    398   0.052
##  3  0.7    398   0.052
##  4  1      398   0.052
##  5  0.1    398 Inf
##  6  0.6    398 Inf
##  7  1      398 Inf
##  8  1.5    398 Inf
```

```
##  9  3      398 Inf
## 10  3      224   0.3
## # ... with 13 more rows
```

Exploratory Data Analysis

```
# We transform the 'Fr' variable using the sigmoid function so that this variable
# will be within a finite range.

train1 <- train %>%
  mutate(Fr_sigmoid = 1 / ( 1 + exp(-Fr))) %>%
  subset(select = c(1:3, 8, 4:7))

train1
```

```
## # A tibble: 89 x 8
##       St    Re      Fr Fr_sigmoid R_moment_1 R_moment_2 R_moment_3 R_moment_4
##    <dbl> <dbl>   <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
##  1  0.1    224   0.052      0.513   0.00216     0.130      14.4       1586.
##  2  3      224   0.052      0.513   0.00379     0.470      69.9      10404
##  3  0.7    224 Inf          1       0.00291     0.0435      0.822       15.6
##  4  0.05    90 Inf          1       0.0635      0.0907      0.467        3.27
##  5  0.7    398 Inf          1       0.000369    0.00622     0.126        2.57
##  6  2       90   0.3        0.574   0.148       2.01       36.2        672.
##  7  0.2     90 Inf          1       0.0813      0.324       3.04        33.0
##  8  3      224 Inf          1       0.00575     0.120       2.75        63.2
##  9  0.9    224 Inf          1       0.00302     0.0452      0.845       15.8
## 10  0.6    398   0.052      0.513   0.000314    0.00447     0.0821       1.51
## # ... with 79 more rows
```

```
cor(train1)
```

```
##                     St          Re  Fr  Fr_sigmoid R_moment_1 R_moment_2
## St           1.00000000 -0.03169871 NaN -0.04734175  0.2147681  0.1479257
## Re          -0.03169871  1.00000000 NaN  0.11152749 -0.7747206 -0.3932344
## Fr                  NaN         NaN   1         NaN        NaN        NaN
## Fr_sigmoid -0.04734175  0.11152749 NaN  1.00000000 -0.1364384 -0.2896720
## R_moment_1  0.21476813 -0.77472058 NaN -0.13643841  1.0000000  0.6298829
## R_moment_2  0.14792571 -0.39323445 NaN -0.28967203  0.6298829  1.0000000
## R_moment_3  0.16474648 -0.38442895 NaN -0.28369640  0.6217326  0.9984335
## R_moment_4  0.18004537 -0.37741773 NaN -0.27852083  0.6150484  0.9946671
##             R_moment_3 R_moment_4
## St           0.1647465  0.1800454
## Re          -0.3844289 -0.3774177
## Fr                 NaN        NaN
## Fr_sigmoid -0.2836964 -0.2785208
## R_moment_1  0.6217326  0.6150484
## R_moment_2  0.9984335  0.9946671
## R_moment_3  1.0000000  0.9988414
## R_moment_4  0.9988414  1.0000000
```

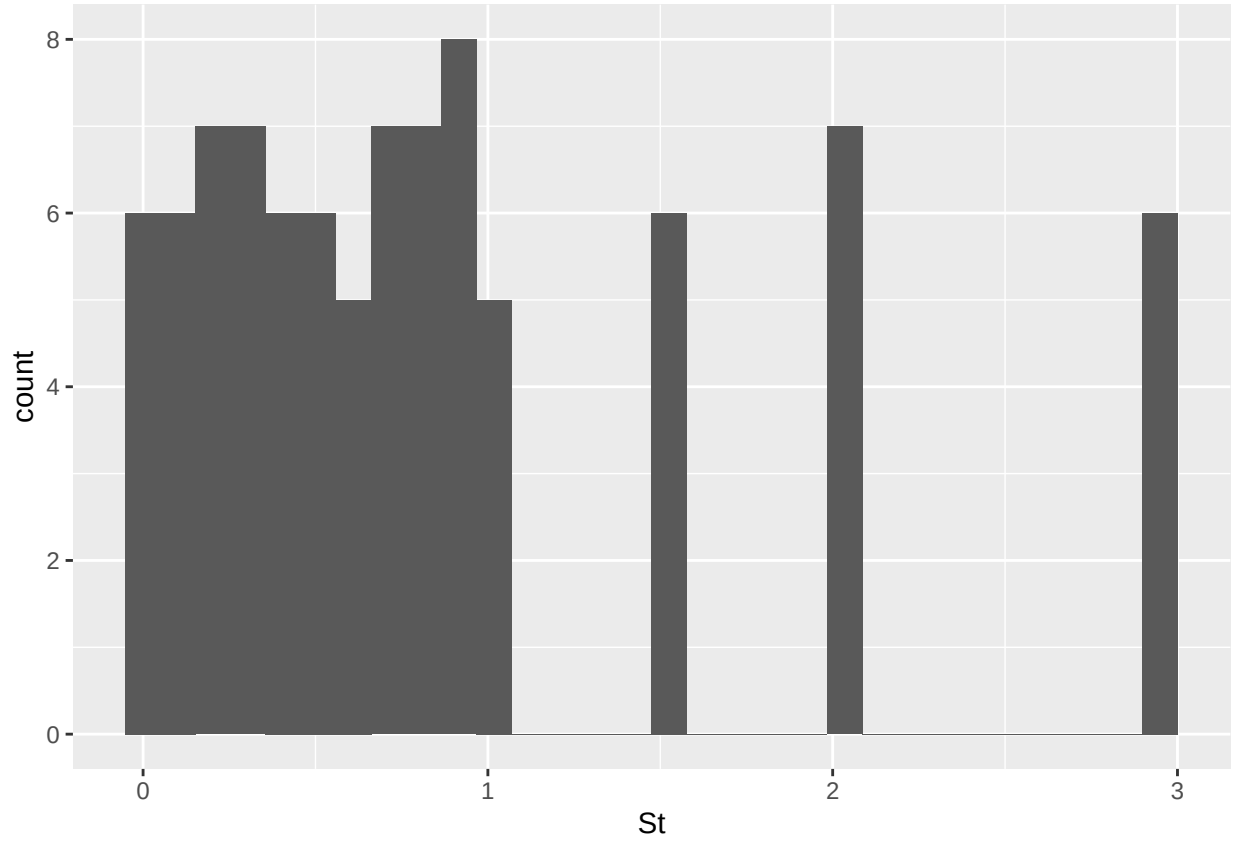
```
test1 <- test %>%
  mutate(Fr_sigmoid = 1 / ( 1 + exp(-Fr)))

test1
```

```
## # A tibble: 23 x 4
##       St    Re      Fr Fr_sigmoid
##    <dbl> <dbl>   <dbl>      <dbl>
##  1  0.05   398   0.052      0.513
##  2  0.2    398   0.052      0.513
##  3  0.7    398   0.052      0.513
##  4  1      398   0.052      0.513
##  5  0.1    398 Inf          1
##  6  0.6    398 Inf          1
##  7  1      398 Inf          1
##  8  1.5    398 Inf          1
##  9  3      398 Inf          1
## 10  3      224   0.3        0.574
## # ... with 13 more rows
```

```
# R_moment_2 is almost perfectly correlated with R_moment_3 and R_moment 4.
```

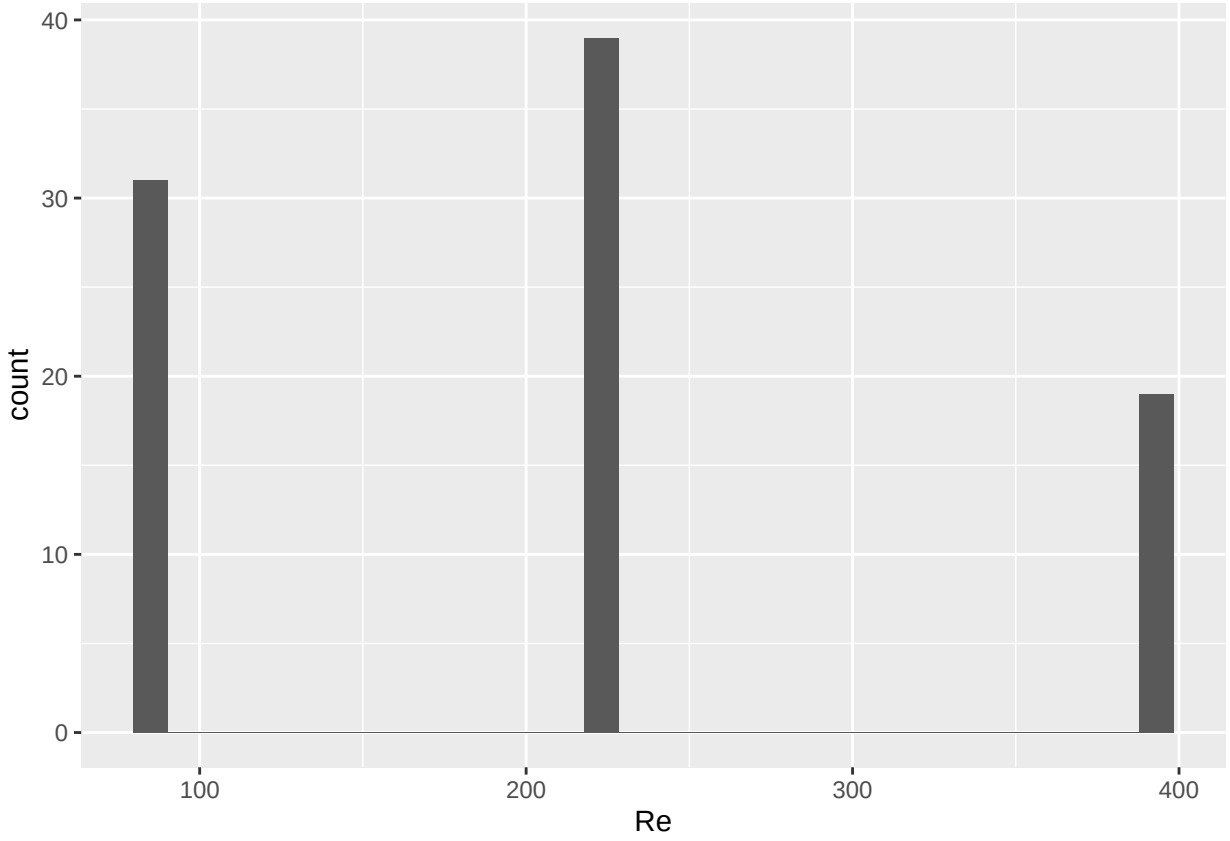
```
ggplot(data = train1, mapping = aes(x = St)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



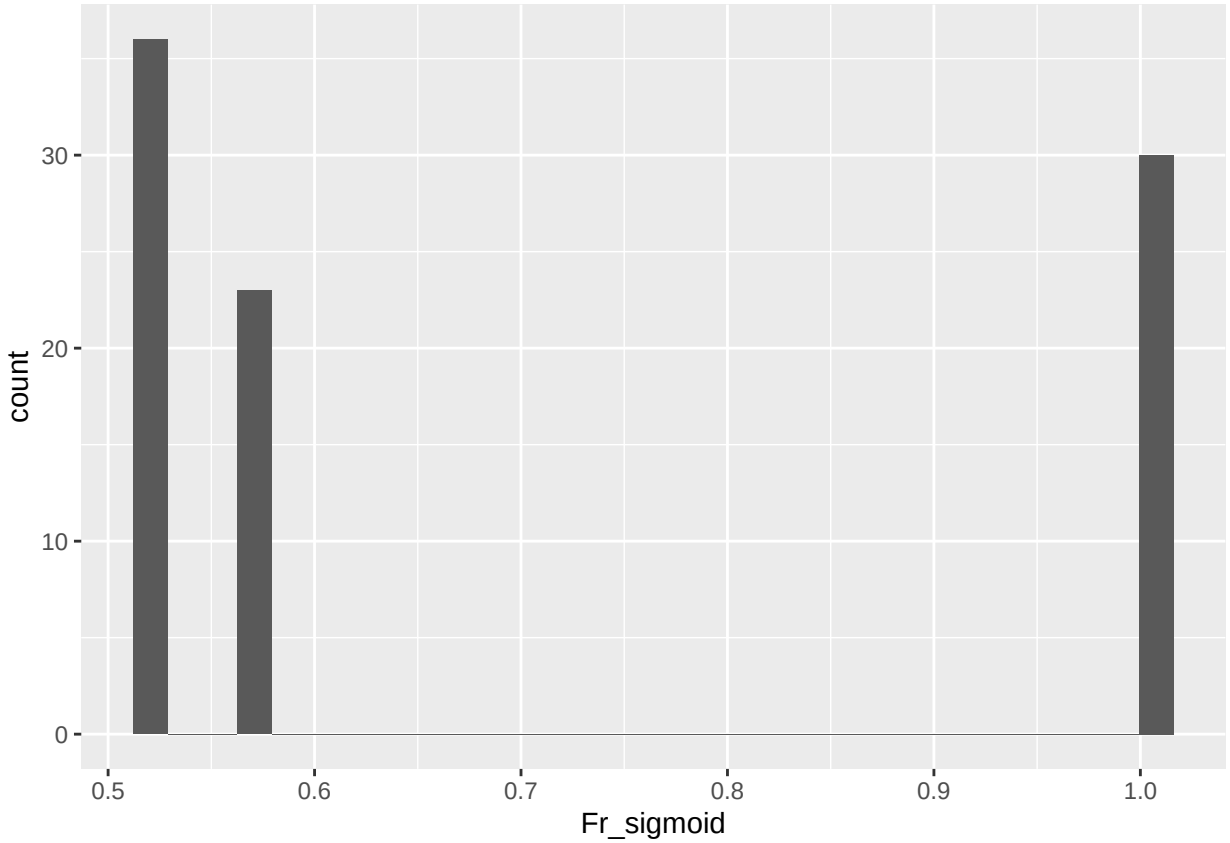
```
ggplot(data = train1, mapping = aes(x = Re)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



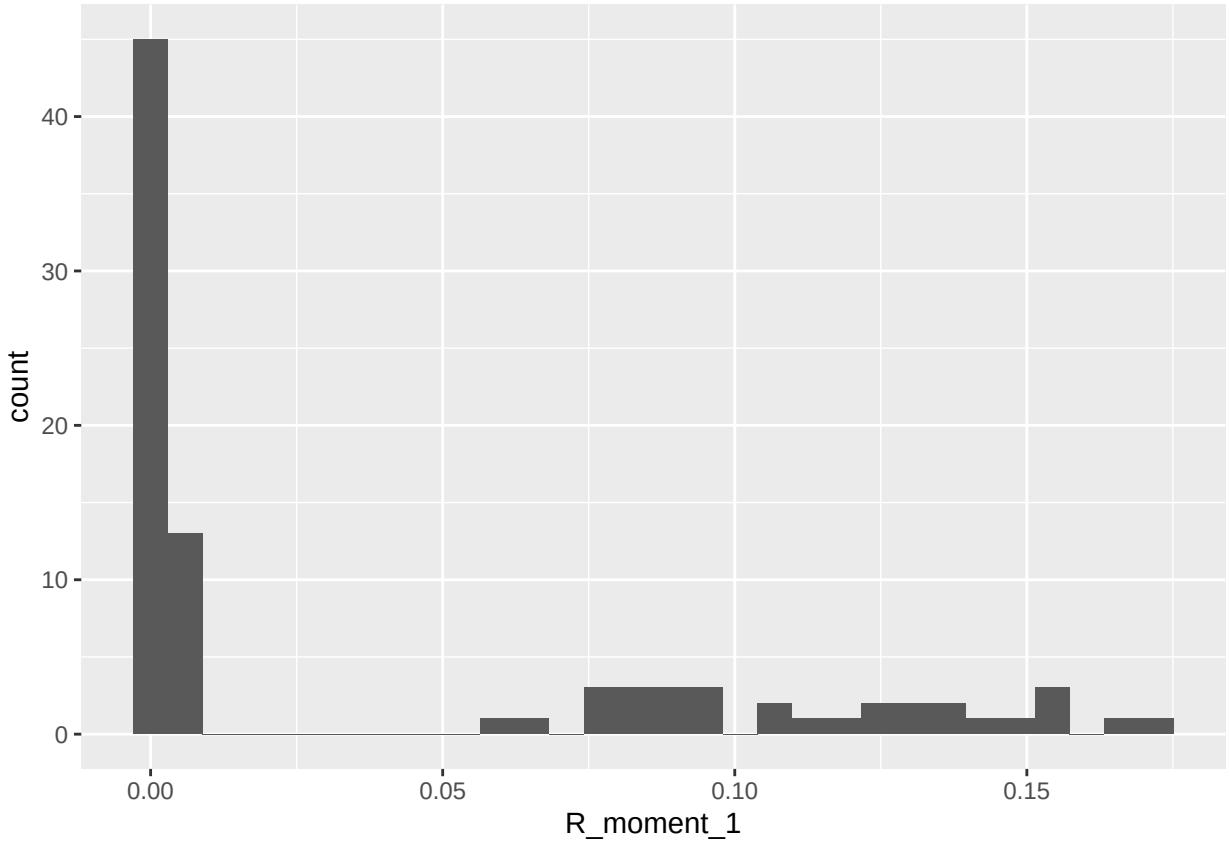

```
ggplot(data = train1, mapping = aes(x = Fr_sigmoid)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



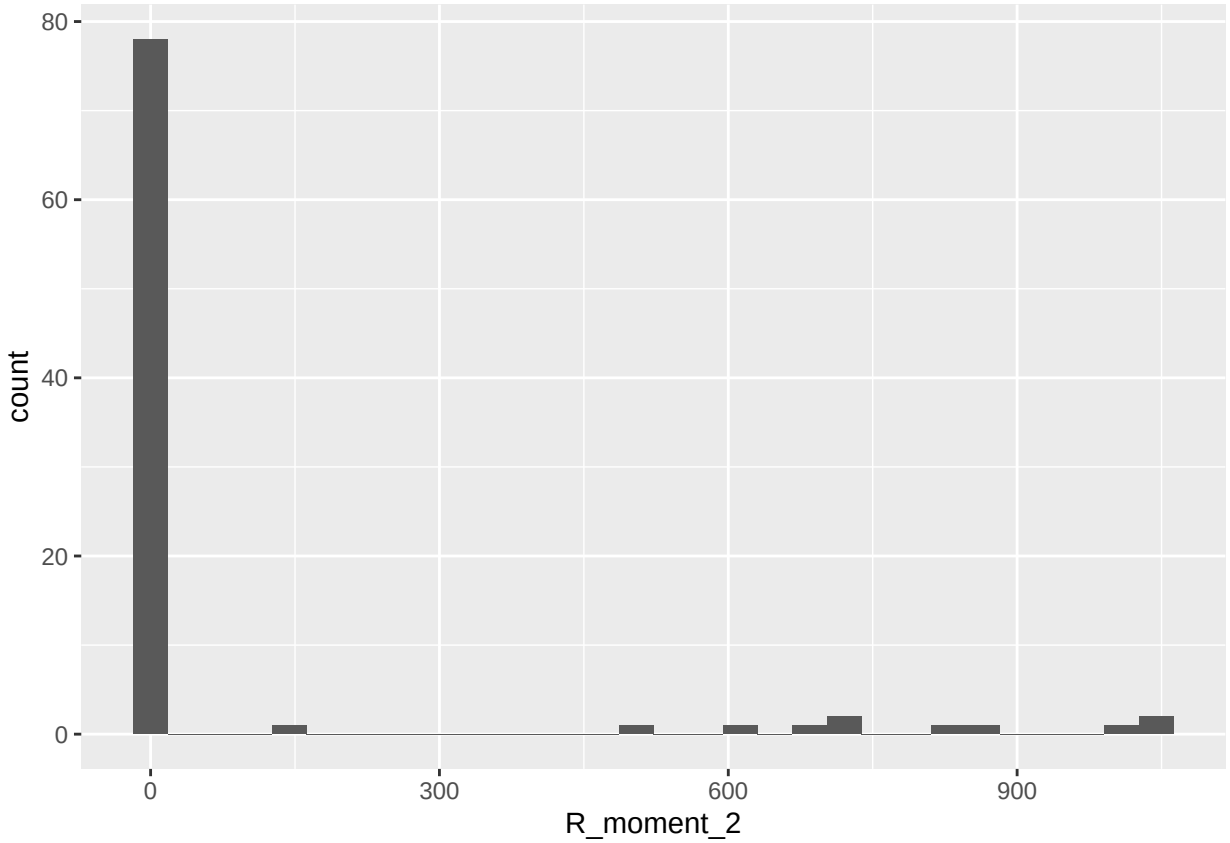

```
ggplot(data = train1, mapping = aes(x = R_moment_1)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



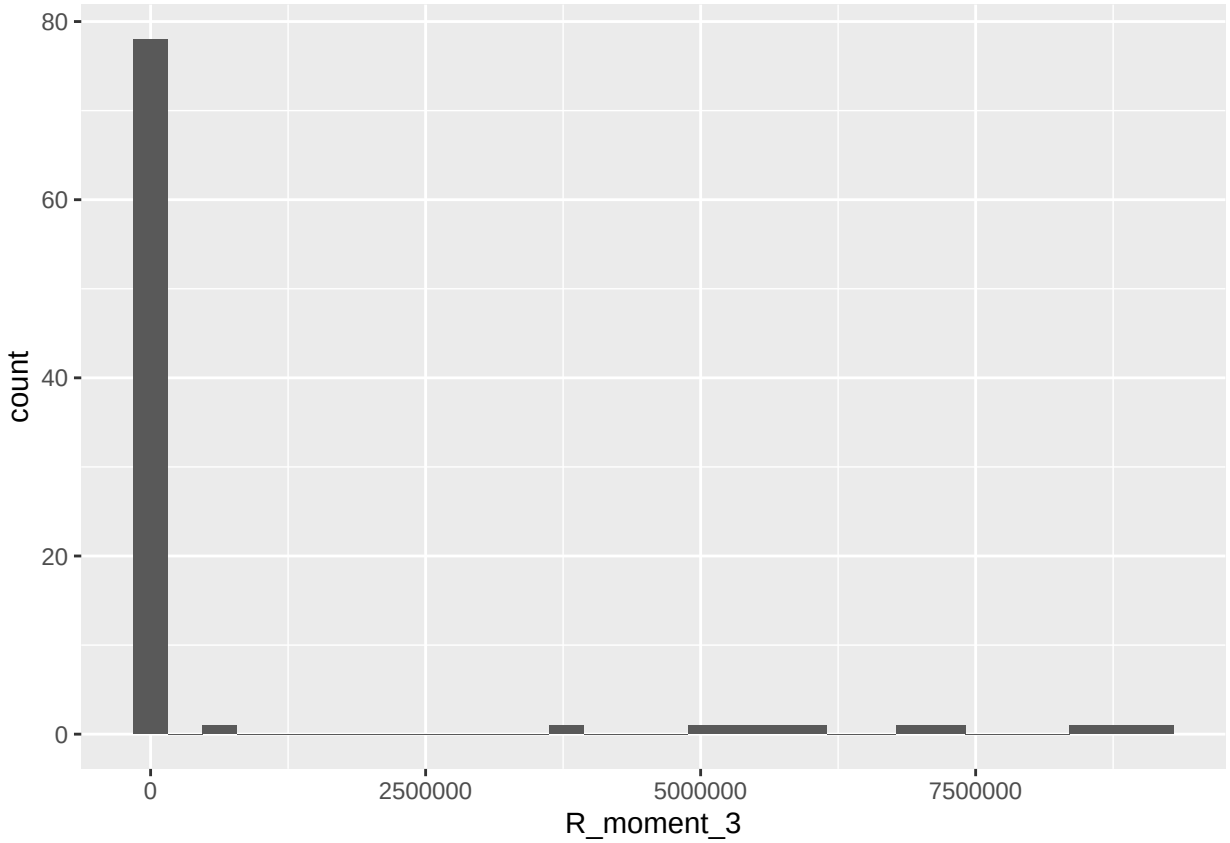

```
ggplot(data = train1, mapping = aes(x = R_moment_2)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



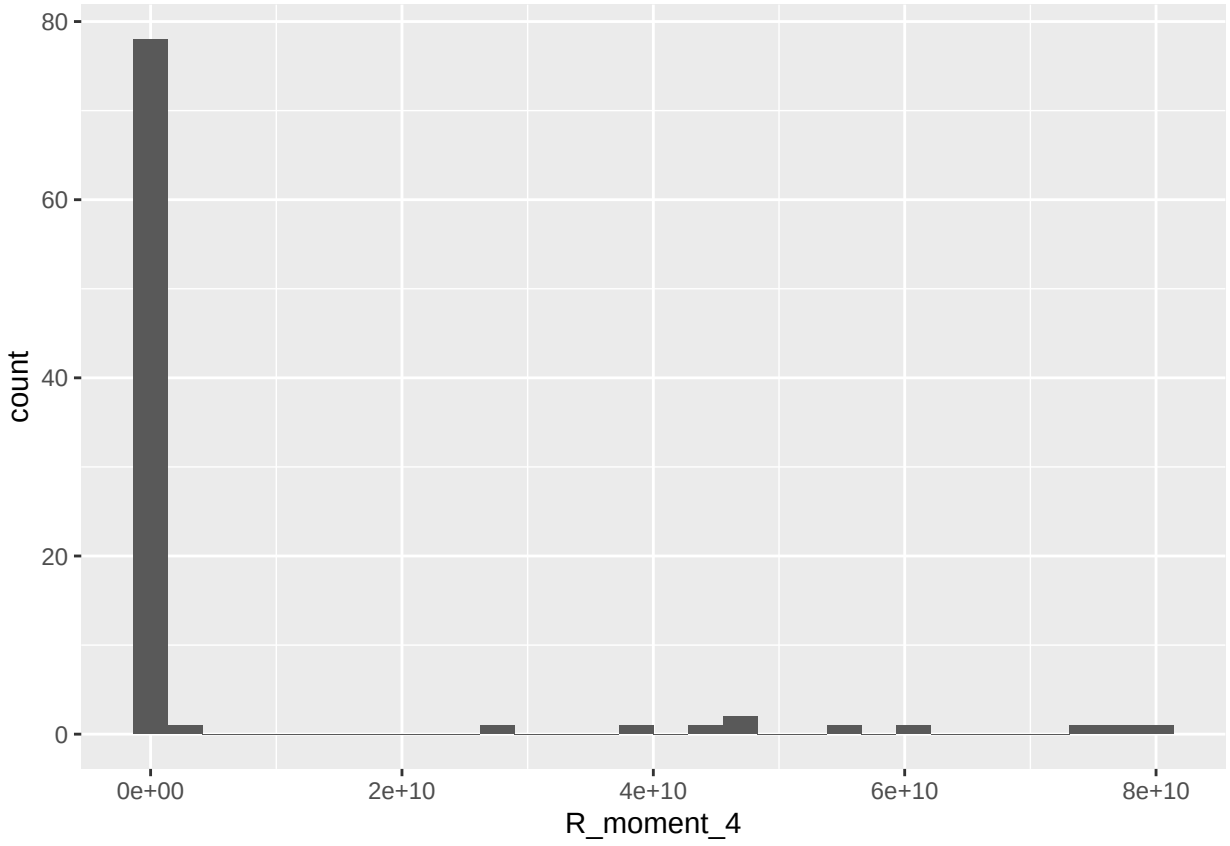

```
ggplot(data = train1, mapping = aes(x = R_moment_3)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

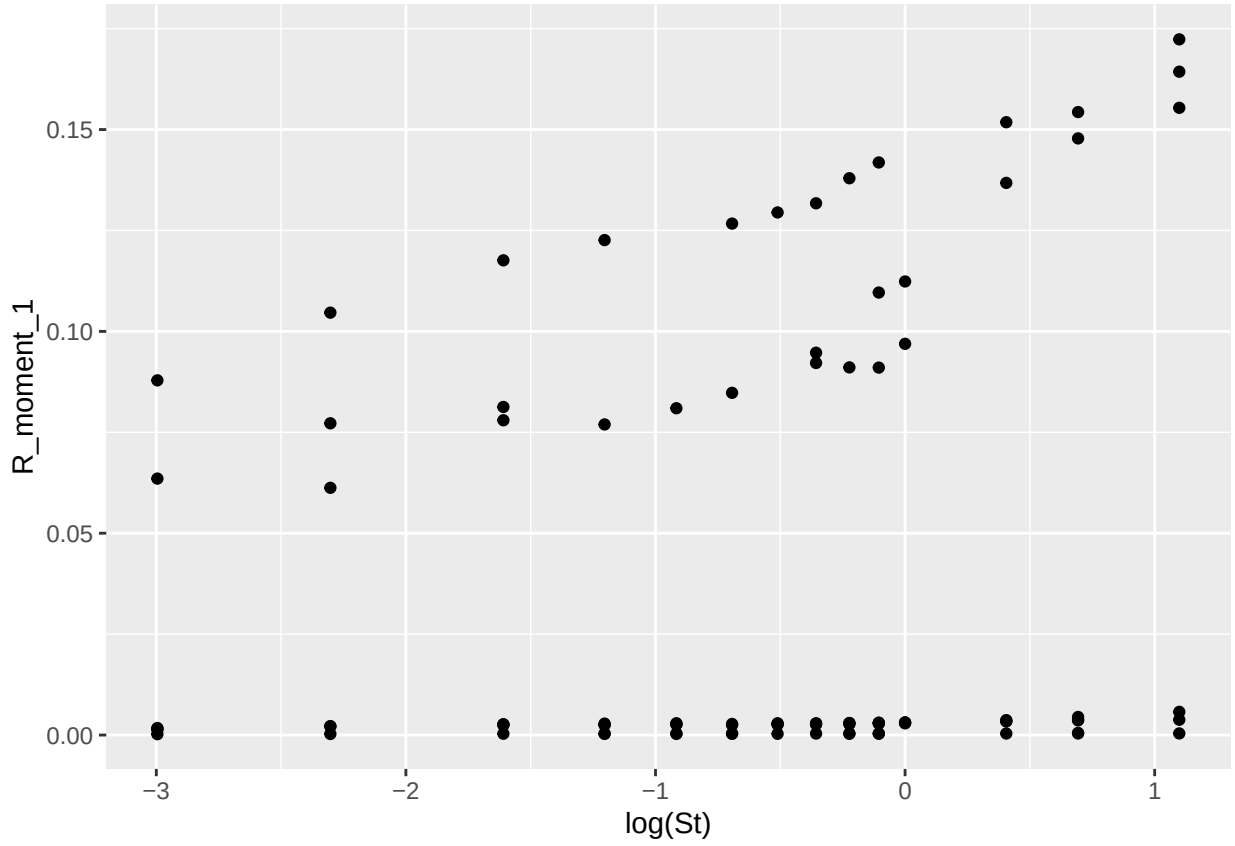



```
ggplot(data = train1, mapping = aes(x = R_moment_4)) + geom_histogram()
```

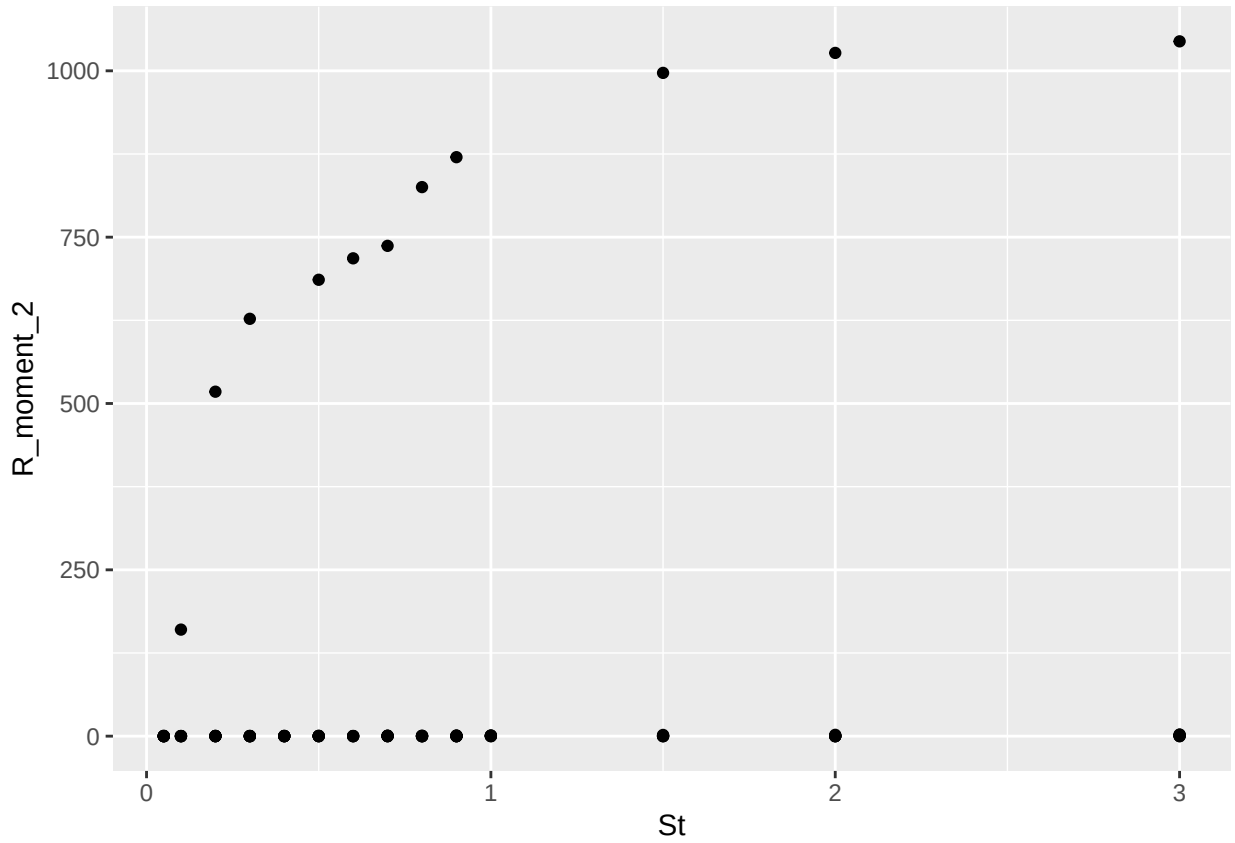
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



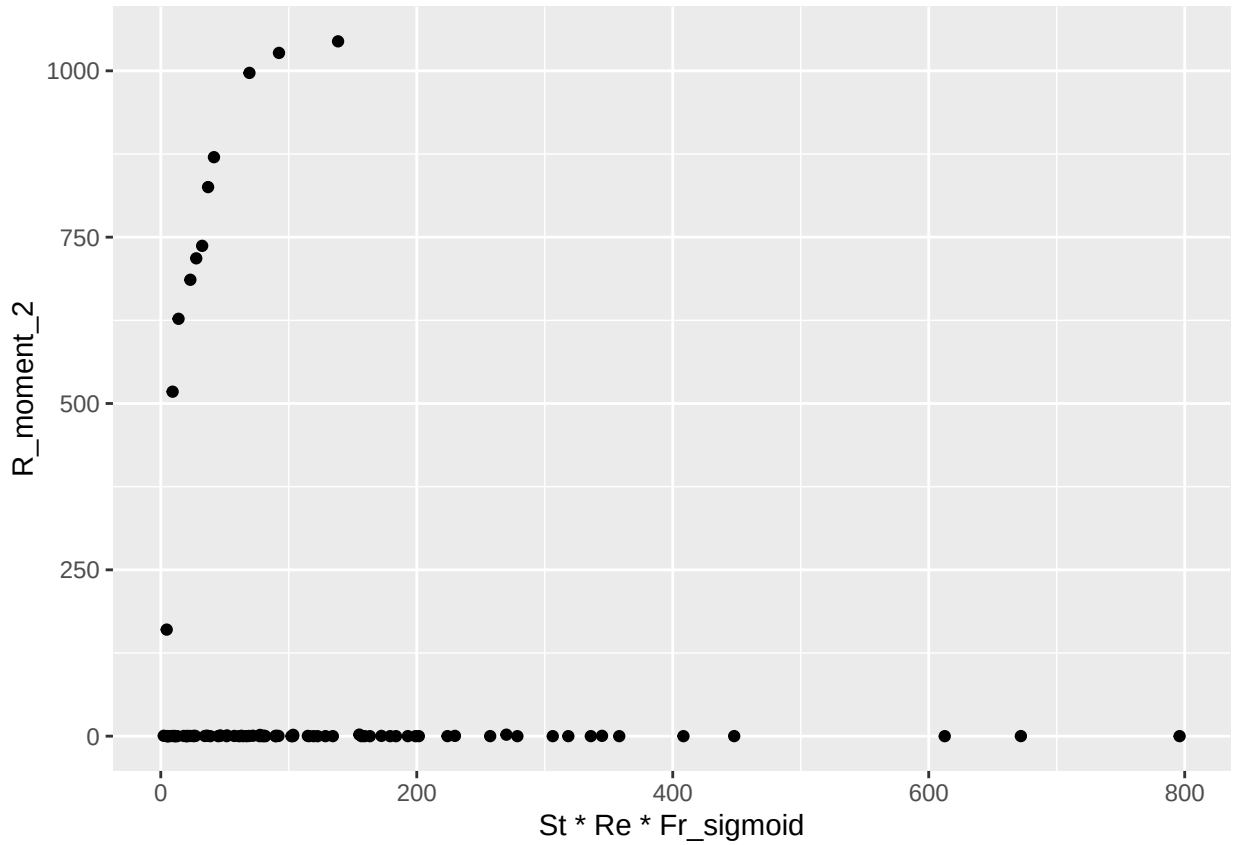

```
ggplot(data = train1, mapping = aes(x = log(St), y = R_moment_1)) + geom_point()
```




```
train1 <- train1 %>%
  mutate(Re_new = case_when(Re == 90 ~ "1",
                            Re == 224 ~ "2",
                            Re == 398 ~ "3"),
         Fr_new = case_when(Fr == 0.052 ~ "1",
                            Fr == 0.3 ~ "2",
                            Fr == Inf ~ "3"))
ggplot(data = train1, mapping = aes(x = St, y = R_moment_2)) + geom_point()
```




```
ggplot(data = train1, mapping = aes(x = St*Re*Fr_sigmoid, y = R_moment_2)) + geom_point()
```



We will try to create these 4 models:

- **Response**: R_moment_1 & **Predictors (Main Effects)**: St, Re, Fr_sigmoid

We will attempt to use a combination of subset selection, polynomial, transformation, and interaction variables.

- **Response**: R_moment_2 & **Predictors (Main Effects)**: St, Re, Fr_sigmoid, R_moment_1

We will attempt to use a combination of subset selection, polynomial, transformation, and interaction variables. We will also include R_moment_1 since it is has significant positive relationship with R_moment_2 (~0.63).

- **Response**: R_moment_3 & **Predictors (Main Effects)**: R_moment_2

We know that R_moment_2 is almost perfectly correlated (>0.99) with R_moment_3, so only using one predictor variable is enough. We try to avoid overfitting by using only R_moment_2 as our only predictor to predict R_moment_3. We will attempt to use polynomial and transformation variables.

- **Response**: R_moment_4 & **Predictors (Main Effects)**: R_moment_2, R_moment_3

Same reasoning - R_moment_2 and R_moment_3 are almost perfectly correlated with R_moment_4. We will only use these 2 predictors and will attempt to use both transformation and interaction variables (since R_moment_2 and R_moment_3 are also highly correlated to each other).

Predictive models

```
# Model 1 (linear)
model1 <- lm(R_moment_1 ~ St + Re_new + Fr_new, data = train1)
summary(model1)
```

```
##
## Call:
## lm(formula = R_moment_1 ~ St + Re_new + Fr_new, data = train1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.038834 -0.008614  0.001702  0.009854  0.039423
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.106488   0.003971  26.818  < 2e-16 ***
## St           0.012213   0.002078   5.877 8.42e-08 ***
## Re_new2     -0.108091   0.003682 -29.353  < 2e-16 ***
## Re_new3     -0.111553   0.004632 -24.081  < 2e-16 ***
## Fr_new2     -0.007623   0.004245  -1.796  0.07618 .
## Fr_new3     -0.010210   0.003787  -2.696  0.00849 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01529 on 83 degrees of freedom
## Multiple R-squared:  0.9293, Adjusted R-squared:  0.9251
## F-statistic: 218.2 on 5 and 83 DF,  p-value: < 2.2e-16
```

```
# Model 1 interactions
model1_int1 <- lm(R_moment_1 ~ St + Re_new + Fr_new + St*Re_new + St*Fr_new + Re_new*Fr_new, data = tra
summary(model1_int1)
```

```
##
## Call:
## lm(formula = R_moment_1 ~ St + Re_new + Fr_new + St * Re_new +
##     St * Fr_new + Re_new * Fr_new, data = train1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0261390 -0.0016048 -0.0000646  0.0024878  0.0140925
##
## Coefficients: (1 not defined because of singularities)
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.108581   0.002378  45.662  < 2e-16 ***
## St               0.024313   0.001744  13.939  < 2e-16 ***
## Re_new2         -0.103060   0.002987 -34.498  < 2e-16 ***
## Re_new3         -0.106710   0.003603 -29.616  < 2e-16 ***
## Fr_new2         -0.035778   0.003540 -10.105 1.05e-15 ***
## Fr_new3         -0.038679   0.003219 -12.015  < 2e-16 ***
## St:Re_new2      -0.027400   0.001938 -14.136  < 2e-16 ***
## St:Re_new3      -0.025834   0.002506 -10.309 4.34e-16 ***
```

```
## St:Fr_new2       0.008944   0.002380   3.759 0.000333 ***
## St:Fr_new3       0.005956   0.001995   2.985 0.003812 **
## Re_new2:Fr_new2  0.028831   0.003657   7.884 1.84e-11 ***
## Re_new3:Fr_new2        NA         NA      NA       NA
## Re_new2:Fr_new3  0.033657   0.003705   9.084 9.21e-14 ***
## Re_new3:Fr_new3  0.034294   0.004051   8.465 1.41e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006253 on 76 degrees of freedom
## Multiple R-squared:  0.9892, Adjusted R-squared:  0.9875
## F-statistic: 578.6 on 12 and 76 DF,  p-value: < 2.2e-16
```

```
# Model 2 (linear)
model2 <- lm(R_moment_2 ~ St + Re_new + Fr_new, data = train1)
summary(model2)
```

```
##
## Call:
## lm(formula = R_moment_2 ~ St + Re_new + Fr_new, data = train1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -383.34 -149.69  -76.02  117.90  575.19
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   382.14      51.90   7.363 1.19e-10 ***
## St             34.79      27.16   1.281    0.204
## Re_new2      -256.24      48.13  -5.324 8.51e-07 ***
## Re_new3      -309.59      60.55  -5.113 2.00e-06 ***
## Fr_new2      -266.10      55.49  -4.796 7.04e-06 ***
## Fr_new3      -214.79      49.50  -4.339 4.00e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 199.8 on 83 degrees of freedom
## Multiple R-squared:  0.4506, Adjusted R-squared:  0.4175
## F-statistic: 13.61 on 5 and 83 DF,  p-value: 1.068e-09
```

```
# Model 2 interactions
model2_int1 <- lm(R_moment_2 ~ St + Re_new + Fr_new + St*Fr_new, data = train1)
summary(model2_int1)
```

```
##
## Call:
## lm(formula = R_moment_2 ~ St + Re_new + Fr_new + St * Fr_new,
##     data = train1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -357.42 -154.74  -20.24  115.95  521.00
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   331.80      56.79   5.842 1.03e-07 ***
## St             96.07      39.60   2.426   0.0175 *
## Re_new2      -262.12      47.82  -5.481 4.66e-07 ***
## Re_new3      -321.16      59.84  -5.367 7.43e-07 ***
## Fr_new2      -148.59      81.17  -1.831   0.0708 .
## Fr_new3      -129.73      71.78  -1.807   0.0744 .
## St:Fr_new2   -137.18      69.93  -1.962   0.0532 .
## St:Fr_new3    -96.88      61.70  -1.570   0.1203
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 196.7 on 81 degrees of freedom
## Multiple R-squared:  0.4807, Adjusted R-squared:  0.4358
## F-statistic: 10.71 on 7 and 81 DF,  p-value: 1.778e-09
```

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
vif(model2_int1)
```

```
##               GVIF Df GVIF^(1/(2*Df))
## St        2.205522  1        1.485100
## Re_new    1.142126  2        1.033781
## Fr_new    5.368514  2        1.522171
## St:Fr_new 7.652714  2        1.663236
```

Apply to test data