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 variables using the sigmoid function so that this variable
# will be within a finite range.

train1 <- train %>%
  rename(M1 = R_moment_1, M2 = R_moment_2, M3 = R_moment_3, M4 = R_moment_4) %>%
  mutate(Fr_sigmoid = 1 / (1 + exp(-Fr)),
         Re_sigmoid = 1 / (1 + exp(-Re)),
         M1_sigmoid = 1 / (1 + exp(-M1)),
         M2_sigmoid = 1 / (1 + exp(-M2)),
         M3_sigmoid = 1 / (1 + exp(-M3)),
         M4_sigmoid = 1 / (1 + exp(-M4)))

train1
```

```
## # A tibble: 89 x 13
##       St    Re      Fr       M1      M2      M3       M4 Fr_sigmoid Re_sigmoid
##    <dbl> <dbl>   <dbl>    <dbl>   <dbl>   <dbl>    <dbl>      <dbl>      <dbl>
##  1  0.1    224   0.052 0.00216  0.130   14.4     1586.        0.513          1
##  2  3      224   0.052 0.00379  0.470   69.9    10404         0.513          1
##  3  0.7    224 Inf     0.00291  0.0435   0.822     15.6       1              1
##  4  0.05    90 Inf     0.0635   0.0907   0.467      3.27      1              1
##  5  0.7    398 Inf     0.000369 0.00622  0.126      2.57      1              1
##  6  2       90   0.3   0.148    2.01    36.2      672.        0.574          1
##  7  0.2     90 Inf     0.0813   0.324    3.04      33.0       1              1
##  8  3      224 Inf     0.00575  0.120    2.75      63.2       1              1
##  9  0.9    224 Inf     0.00302  0.0452   0.845     15.8       1              1
## 10  0.6    398   0.052 0.000314 0.00447  0.0821     1.51      0.513          1
## # ... with 79 more rows, and 4 more variables: M1_sigmoid <dbl>,
## #   M2_sigmoid <dbl>, M3_sigmoid <dbl>, M4_sigmoid <dbl>
```

```
cor(train1)
```

```
## Warning in cor(train1): the standard deviation is zero
```

```
##                      St          Re  Fr         M1         M2         M3
## 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
## M1          0.21476813 -0.77472058 NaN  1.0000000  0.6298829  0.6217326
## M2          0.14792571 -0.39323445 NaN  0.6298829  1.0000000  0.9984335
## M3          0.16474648 -0.38442895 NaN  0.6217326  0.9984335  1.0000000
## M4          0.18004537 -0.37741773 NaN  0.6150484  0.9946671  0.9988414
## Fr_sigmoid -0.04734175  0.11152749 NaN -0.1364384 -0.2896720 -0.2836964
## Re_sigmoid          NA          NA  NA         NA         NA         NA
## M1_sigmoid  0.21438984 -0.77491487 NaN  0.9999997  0.6297126  0.6215493
```

```
## M2_sigmoid  0.24775970 -0.64993729 NaN  0.8921688  0.8036479  0.7865710
## M3_sigmoid  0.21963439 -0.78152504 NaN  0.6386881  0.3618662  0.3536117
## M4_sigmoid  0.23440506 -0.42010511 NaN  0.2842766  0.1569689  0.1533560
##                    M4   Fr_sigmoid Re_sigmoid M1_sigmoid M2_sigmoid M3_sigmoid
## St          0.1800454 -0.047341748         NA  0.2143898  0.2477597  0.2196344
## Re         -0.3774177  0.111527494         NA -0.7749149 -0.6499373 -0.7815250
## Fr               NaN          NaN         NA        NaN        NaN        NaN
## M1          0.6150484 -0.136438406         NA  0.9999997  0.8921688  0.6386881
## M2          0.9946671 -0.289672032         NA  0.6297126  0.8036479  0.3618662
## M3          0.9988414 -0.283696405         NA  0.6215493  0.7865710  0.3536117
## M4          1.0000000 -0.278520831         NA  0.6148543  0.7722272  0.3471592
## Fr_sigmoid -0.2785208  1.000000000         NA -0.1363550 -0.3129278 -0.2443088
## Re_sigmoid         NA           NA          1         NA         NA         NA
## M1_sigmoid  0.6148543 -0.136354980         NA  1.0000000  0.8920383  0.6387825
## M2_sigmoid  0.7722272 -0.312927783         NA  0.8920383  1.0000000  0.6579482
## M3_sigmoid  0.3471592 -0.244308839         NA  0.6387825  0.6579482  1.0000000
## M4_sigmoid  0.1505576  0.006632018         NA  0.2843257  0.2980432  0.6020816
##              M4_sigmoid
## St          0.234405063
## Re         -0.420105109
## Fr                  NaN
## M1          0.284276600
## M2          0.156968913
## M3          0.153356007
## M4          0.150557597
## Fr_sigmoid  0.006632018
## Re_sigmoid           NA
## M1_sigmoid  0.284325733
## M2_sigmoid  0.298043180
## M3_sigmoid  0.602081557
## M4_sigmoid  1.000000000
```

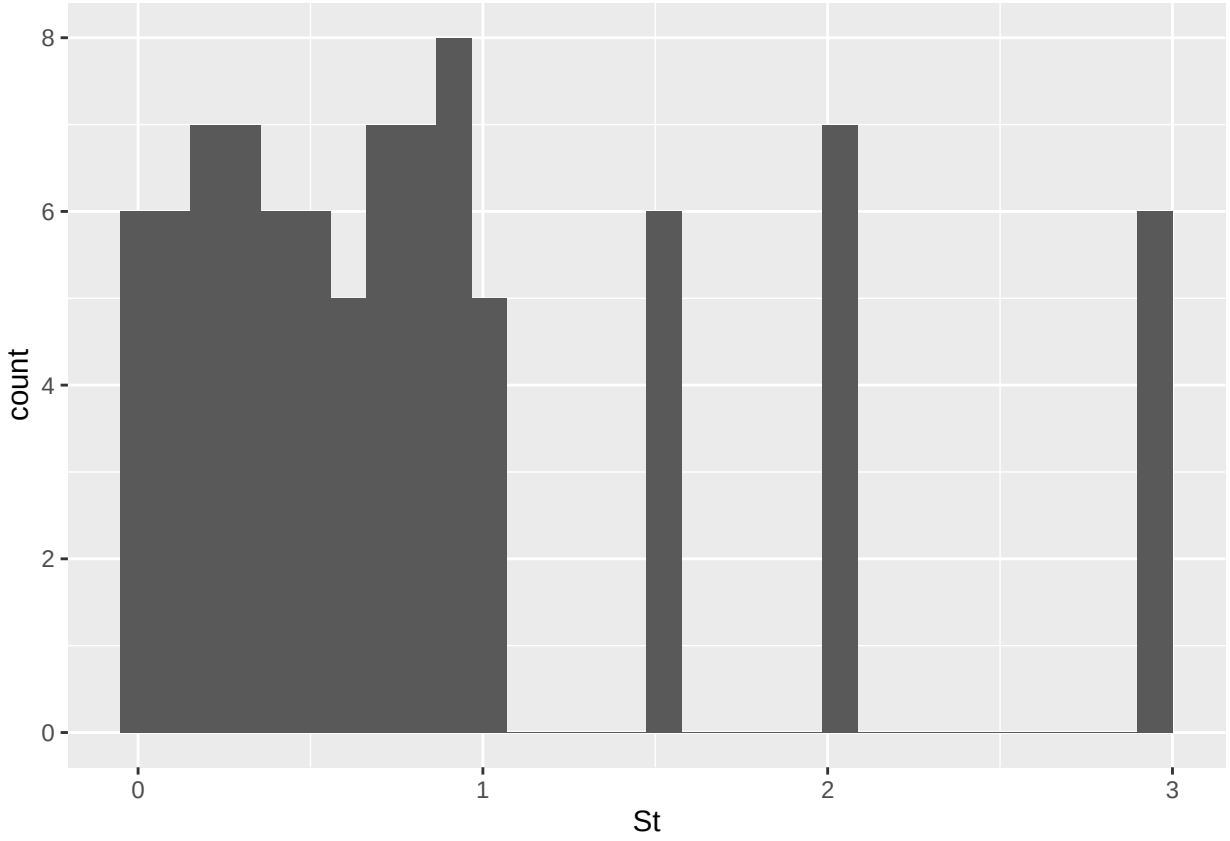
```
test1 <- test %>%
  mutate(Fr_sigmoid = 1 / (1 + exp(-Fr)),
         Re_sigmoid = 1 / (1 + exp(-Re)))

test1
```

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

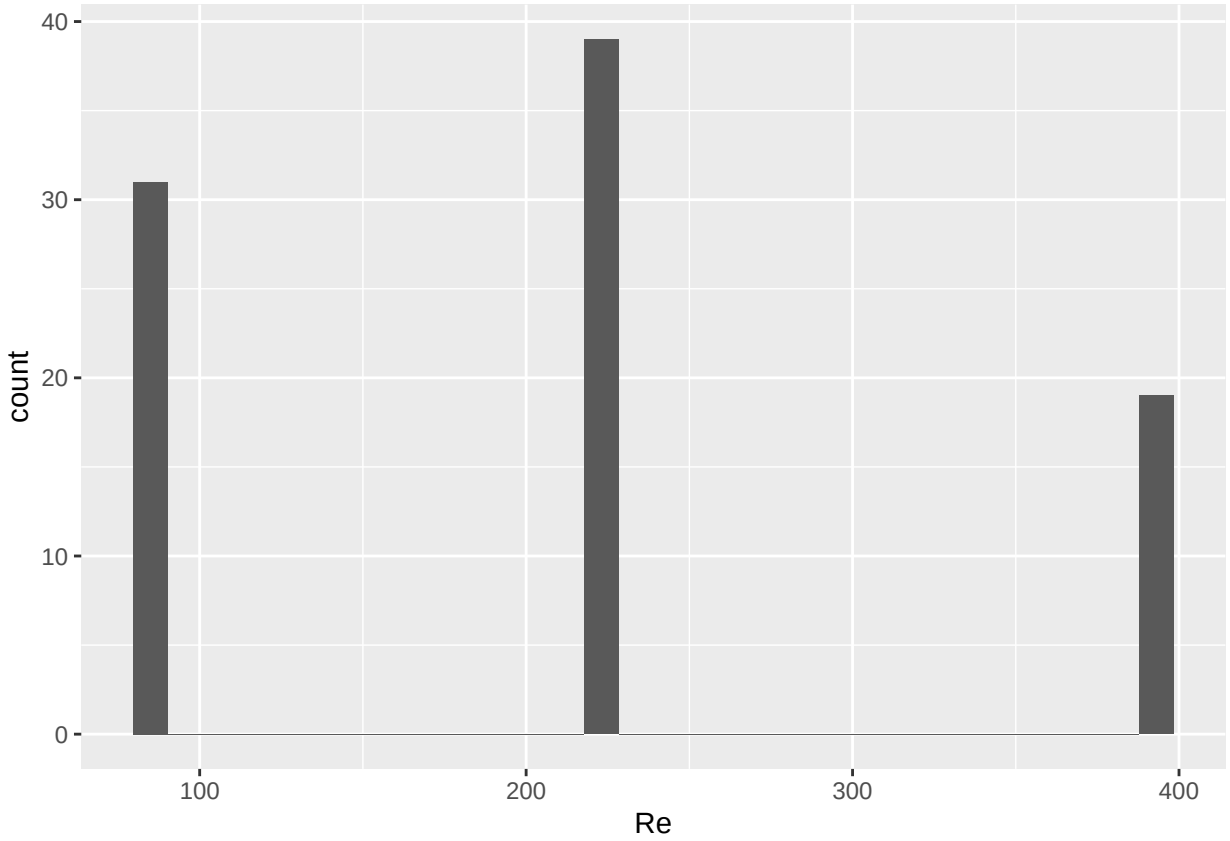
```
ggplot(data = train1, mapping = aes(x = St)) + geom_histogram()
```

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




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

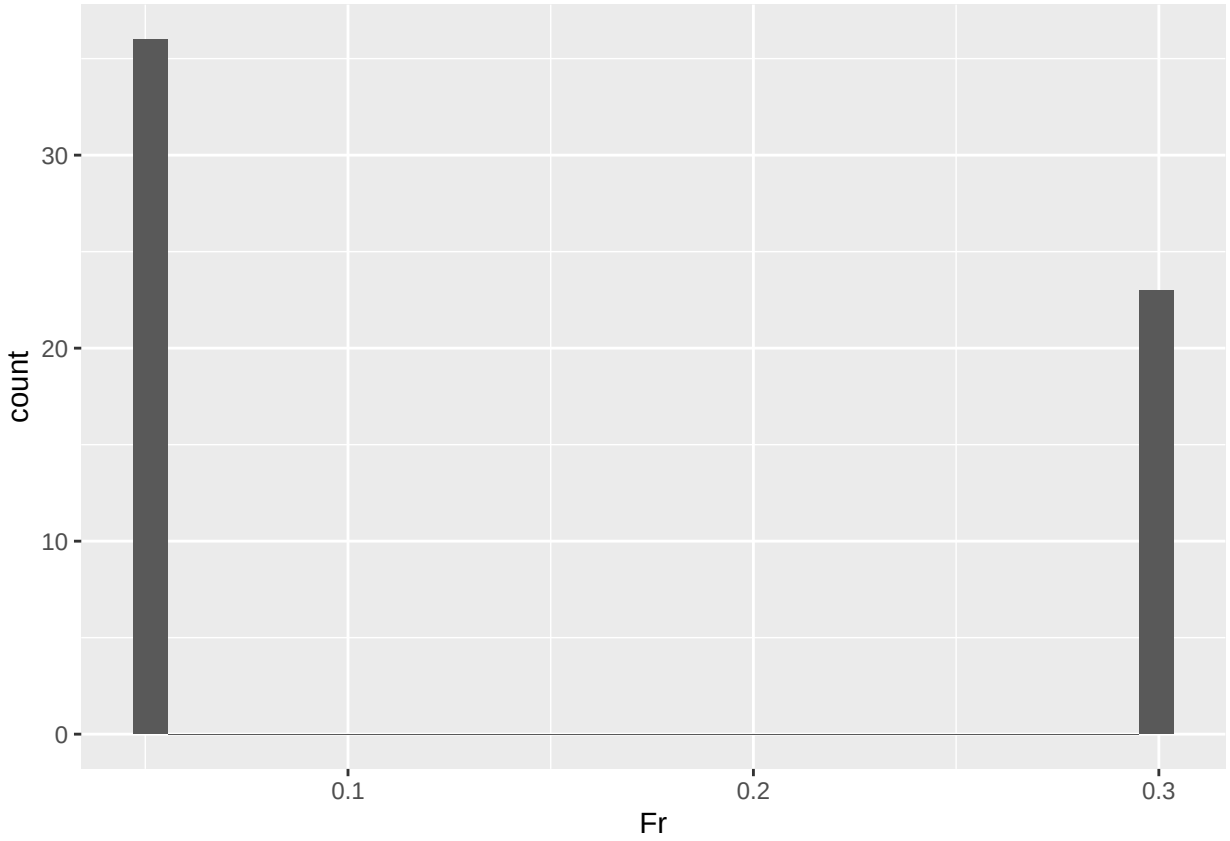
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




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

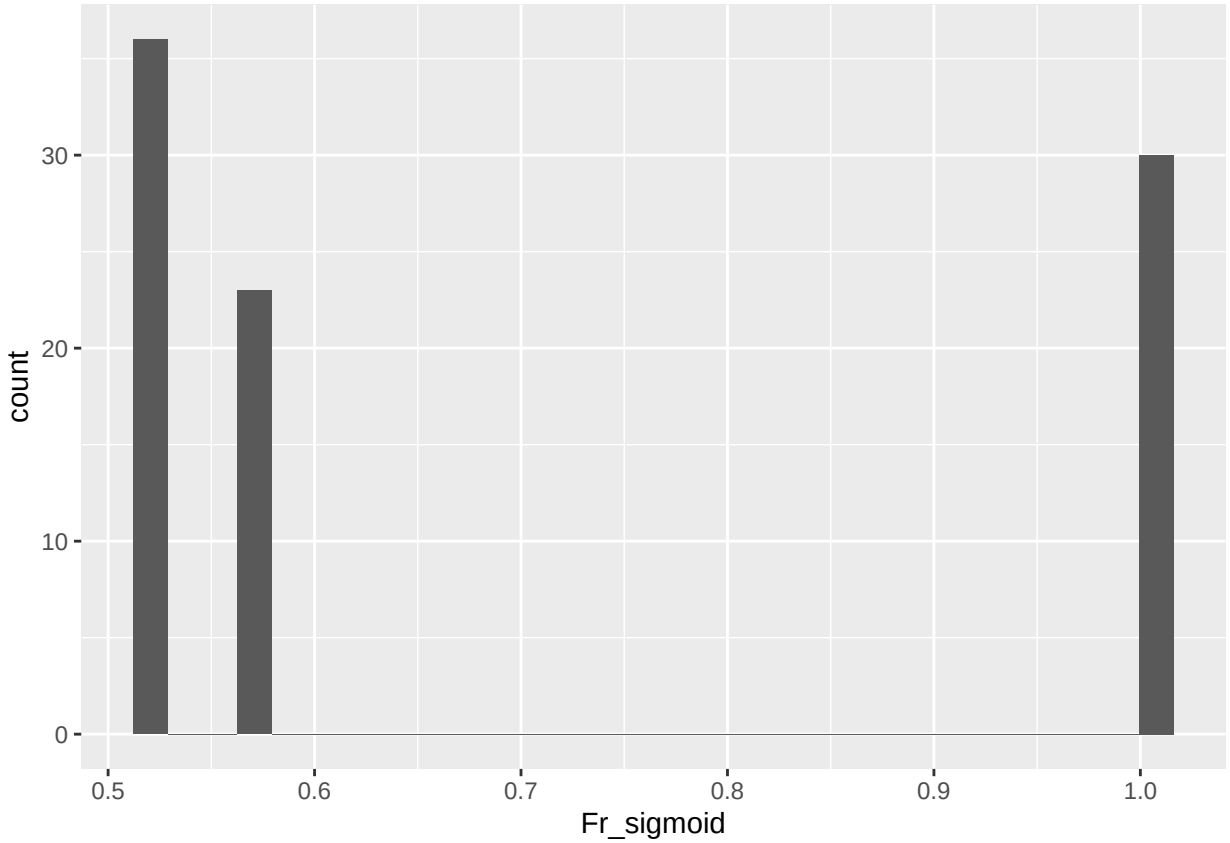
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

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



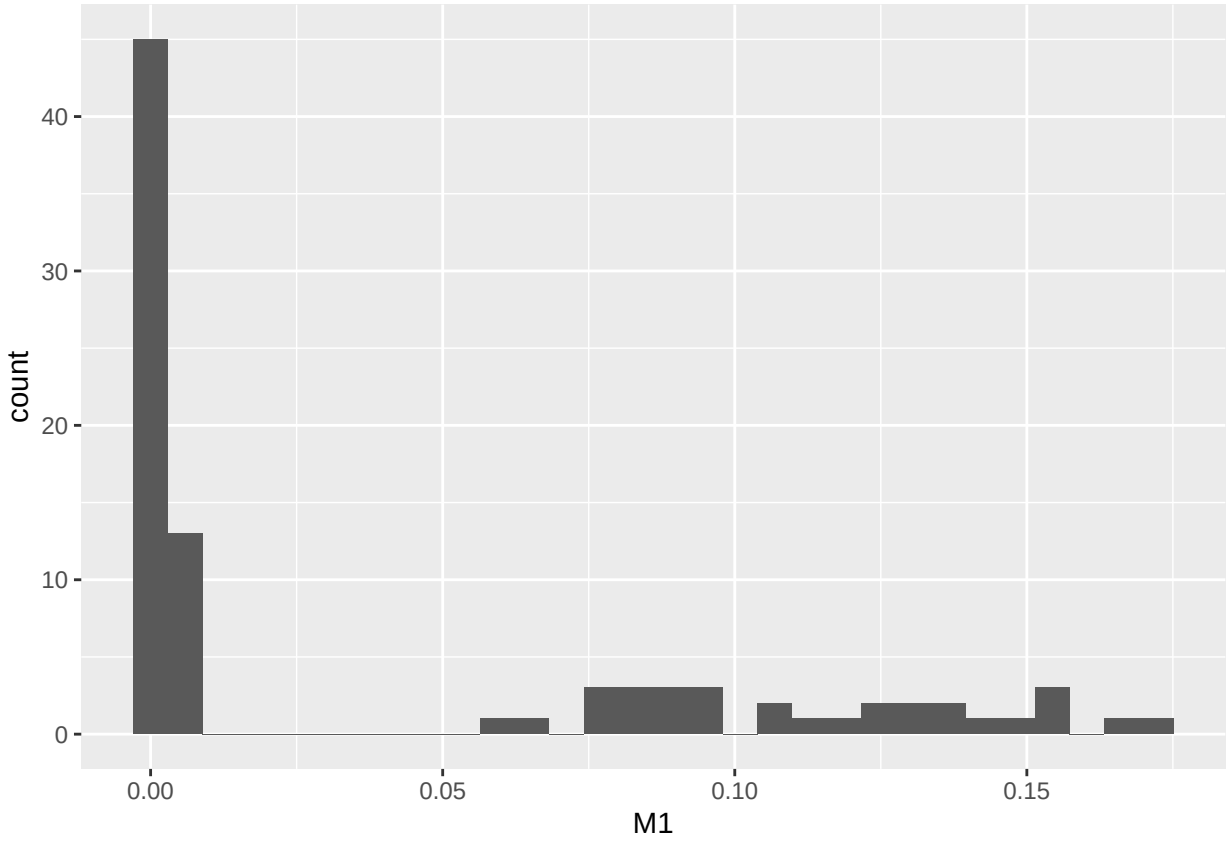

```
ggplot(data = train1, mapping = aes(x = Fr_sigmoid)) + geom_histogram()
```

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



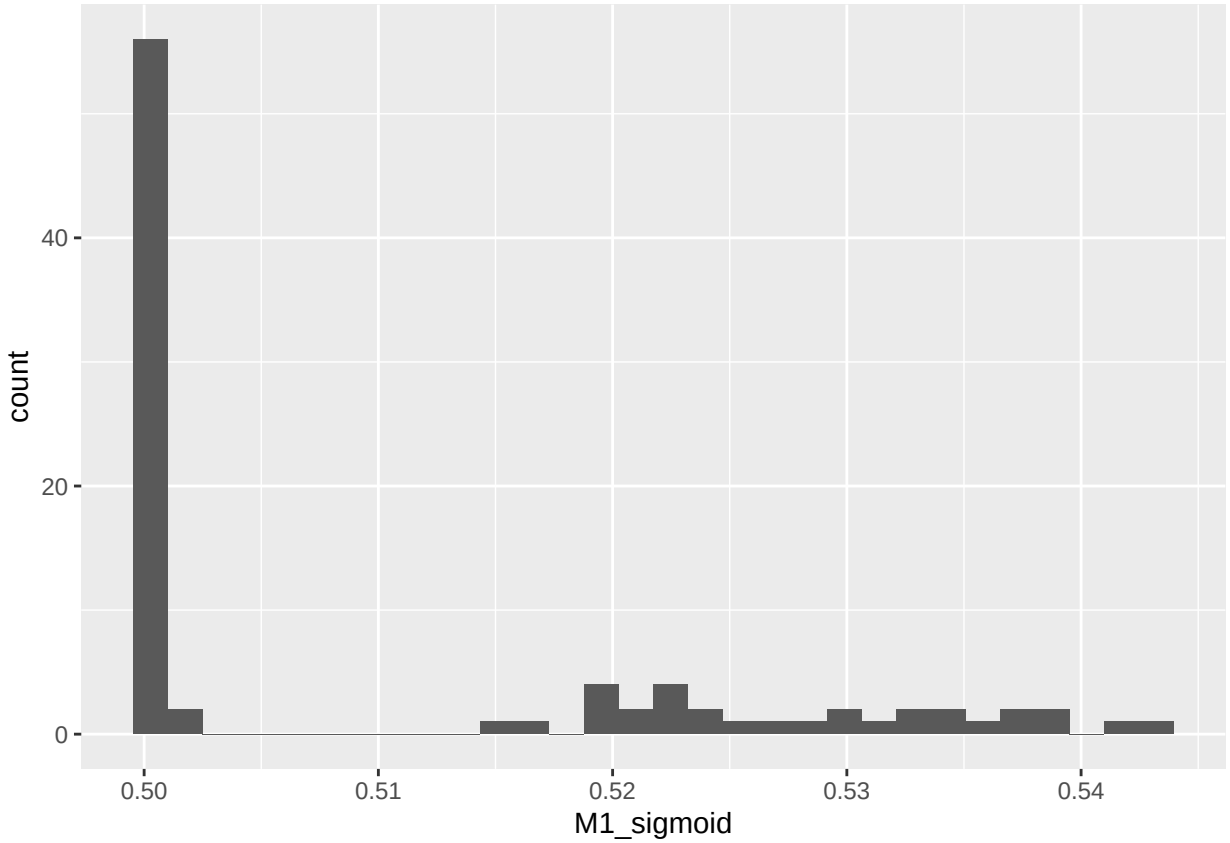

```
ggplot(data = train1, mapping = aes(x = M1)) + geom_histogram()
```

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



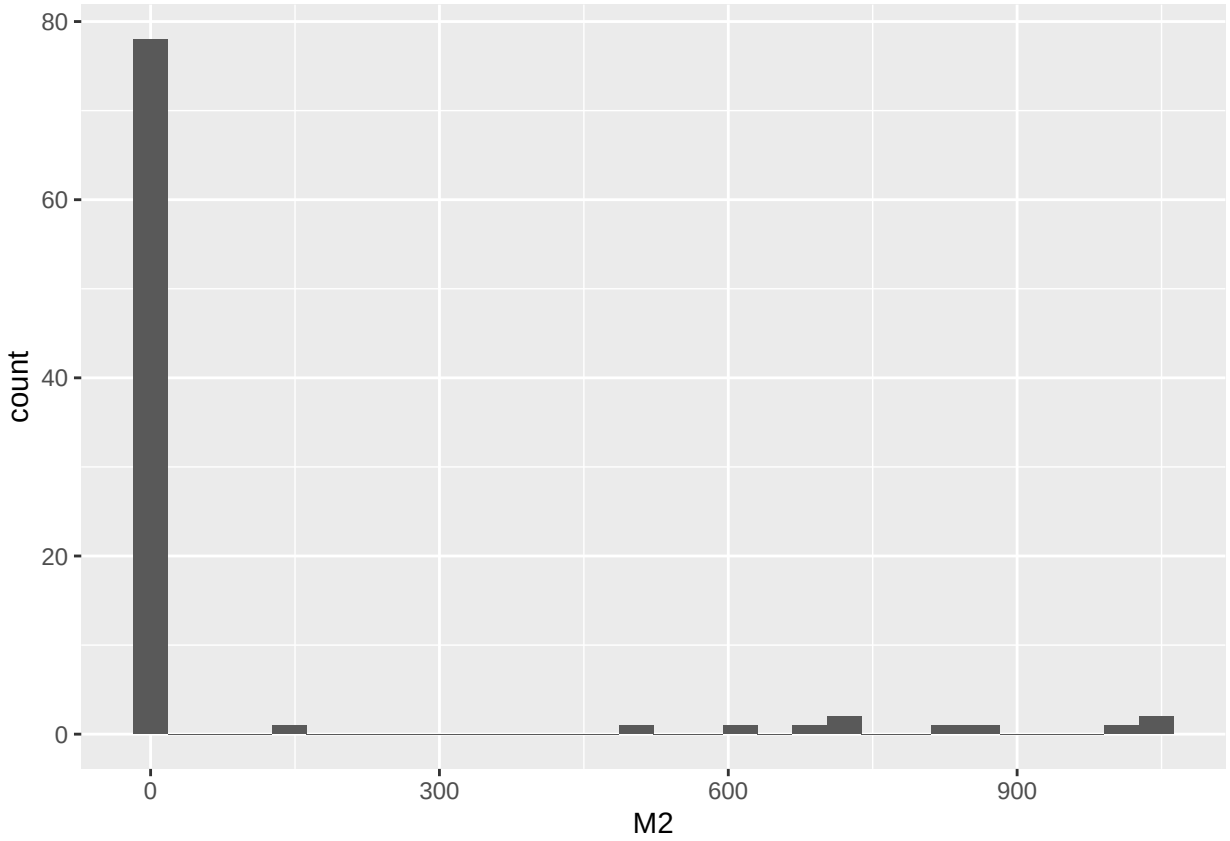

```
ggplot(data = train1, mapping = aes(x = M1_sigmoid)) + geom_histogram()
```

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



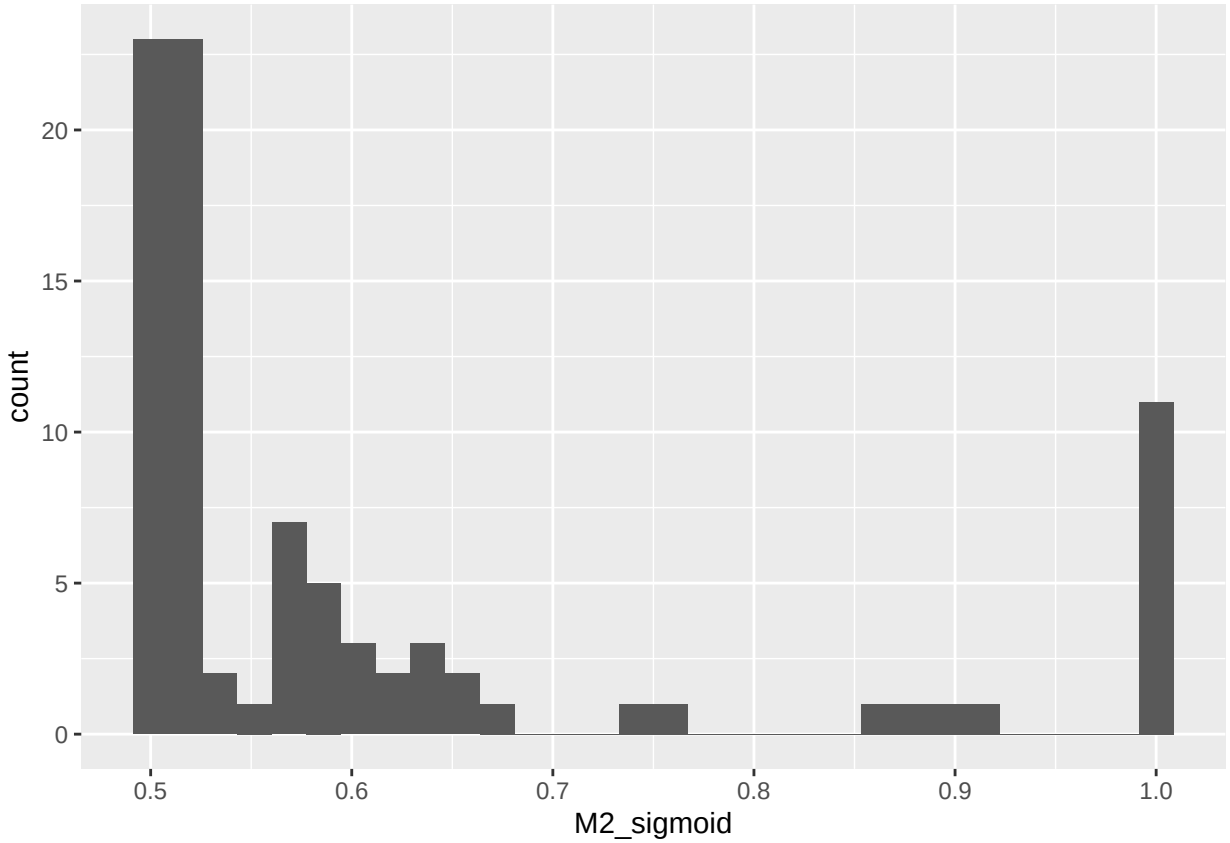

```
ggplot(data = train1, mapping = aes(x = M2)) + geom_histogram()
```

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



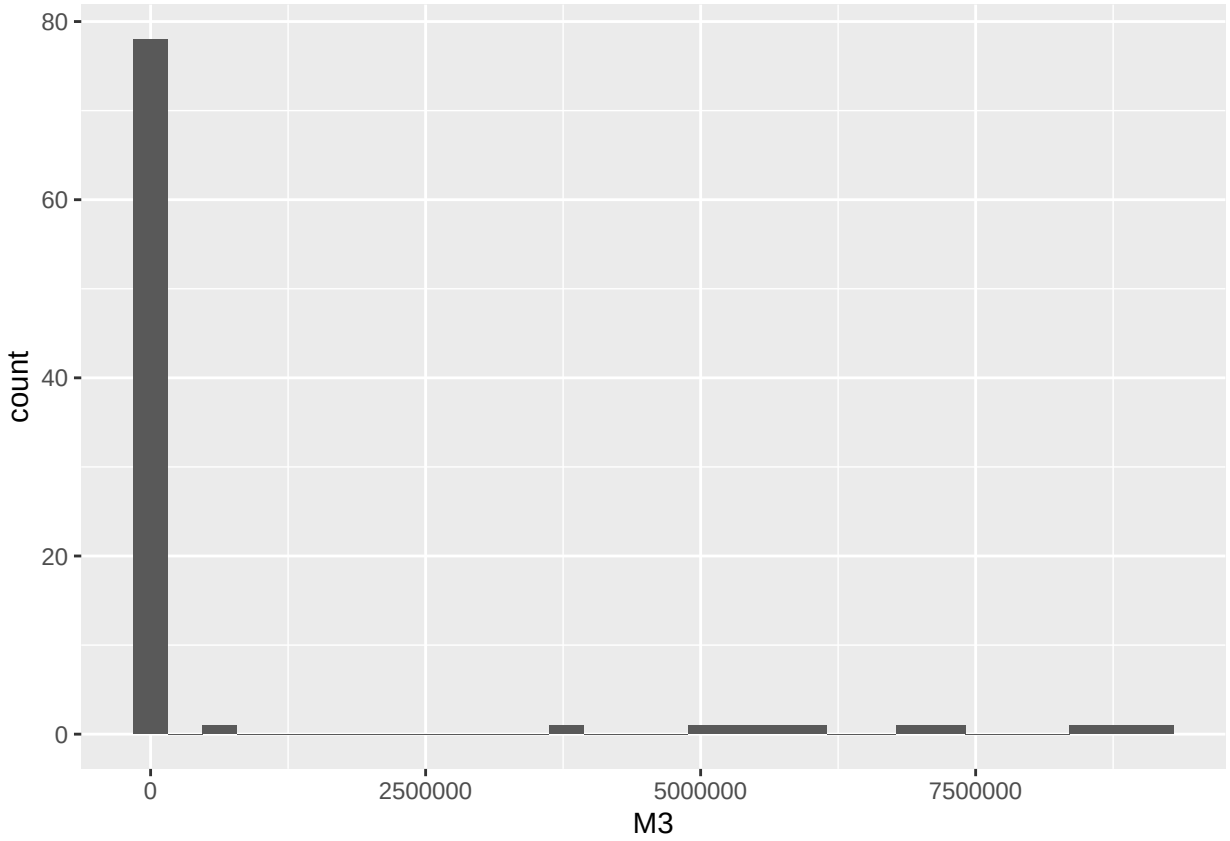

```
ggplot(data = train1, mapping = aes(x = M2_sigmoid)) + geom_histogram()
```

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



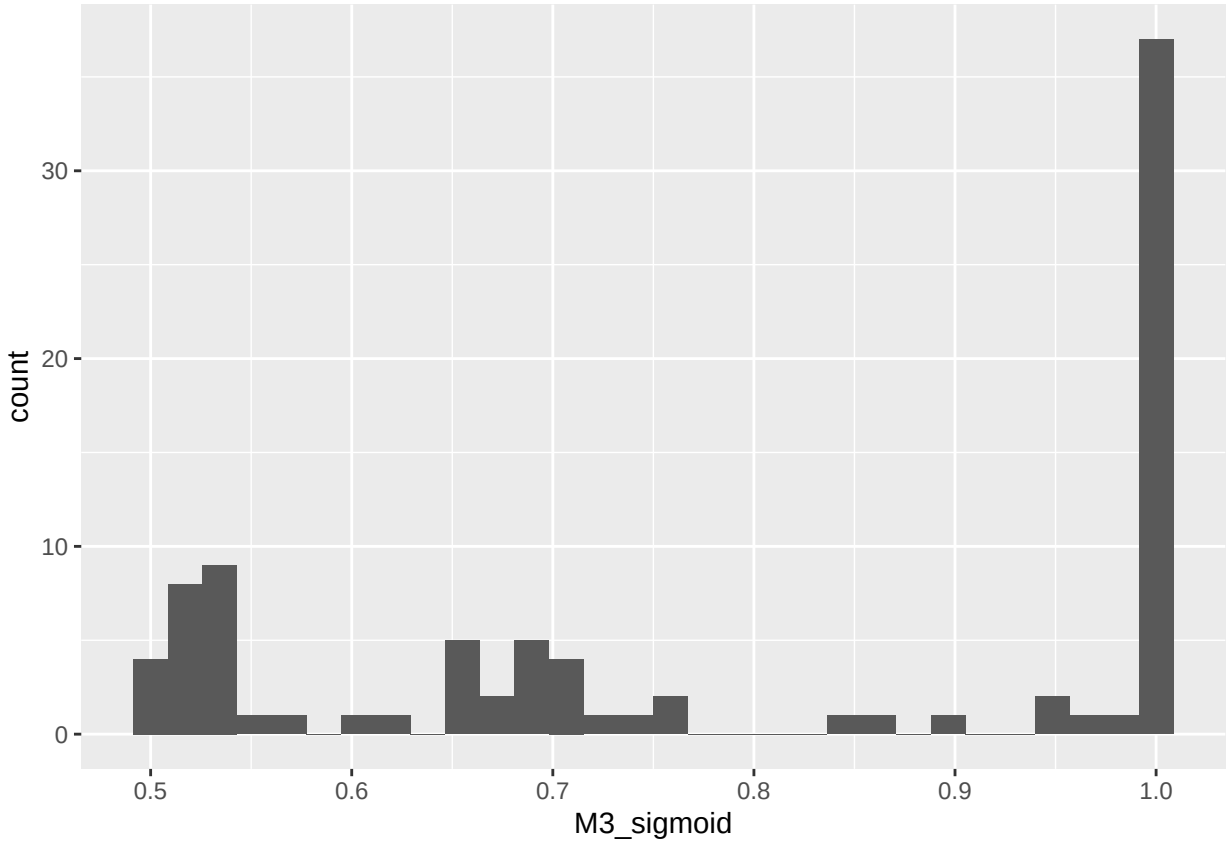

```
ggplot(data = train1, mapping = aes(x = M3)) + geom_histogram()
```

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



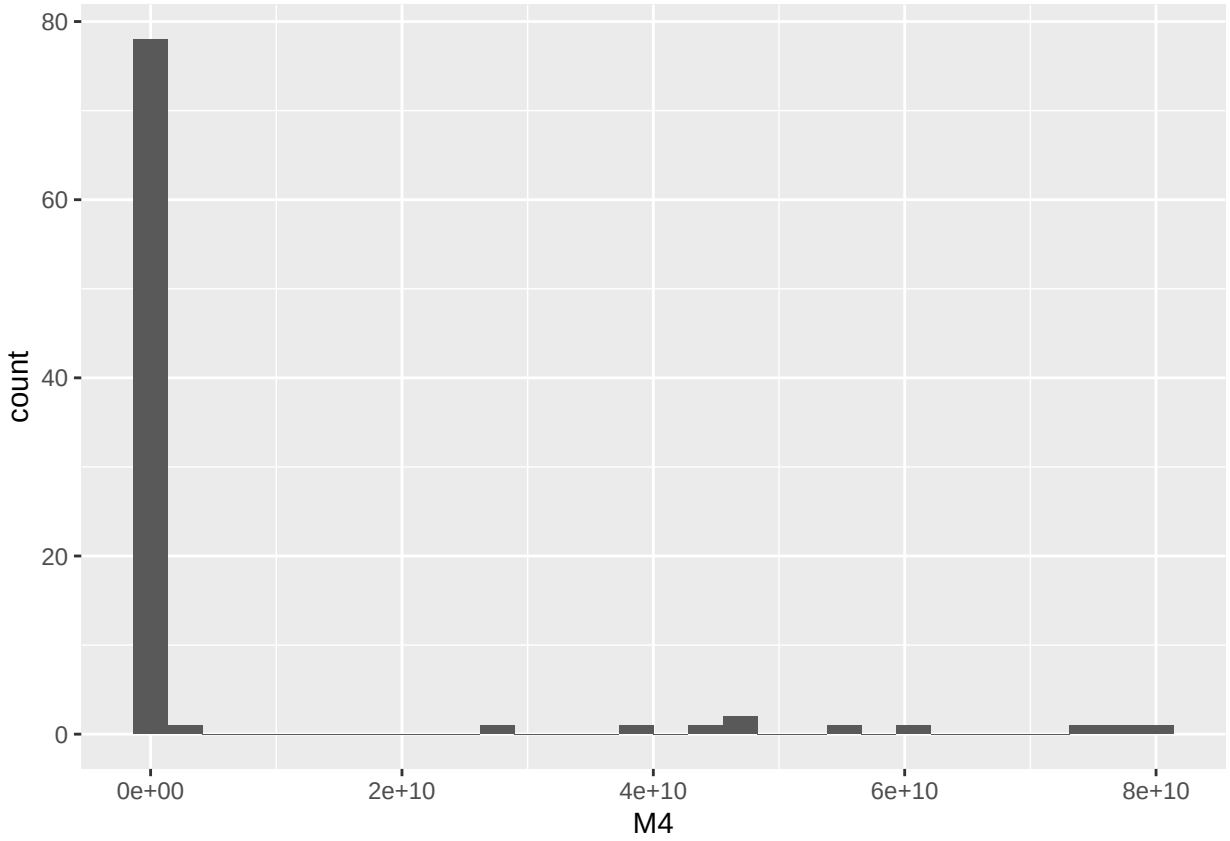

```
ggplot(data = train1, mapping = aes(x = M3_sigmoid)) + geom_histogram()
```

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



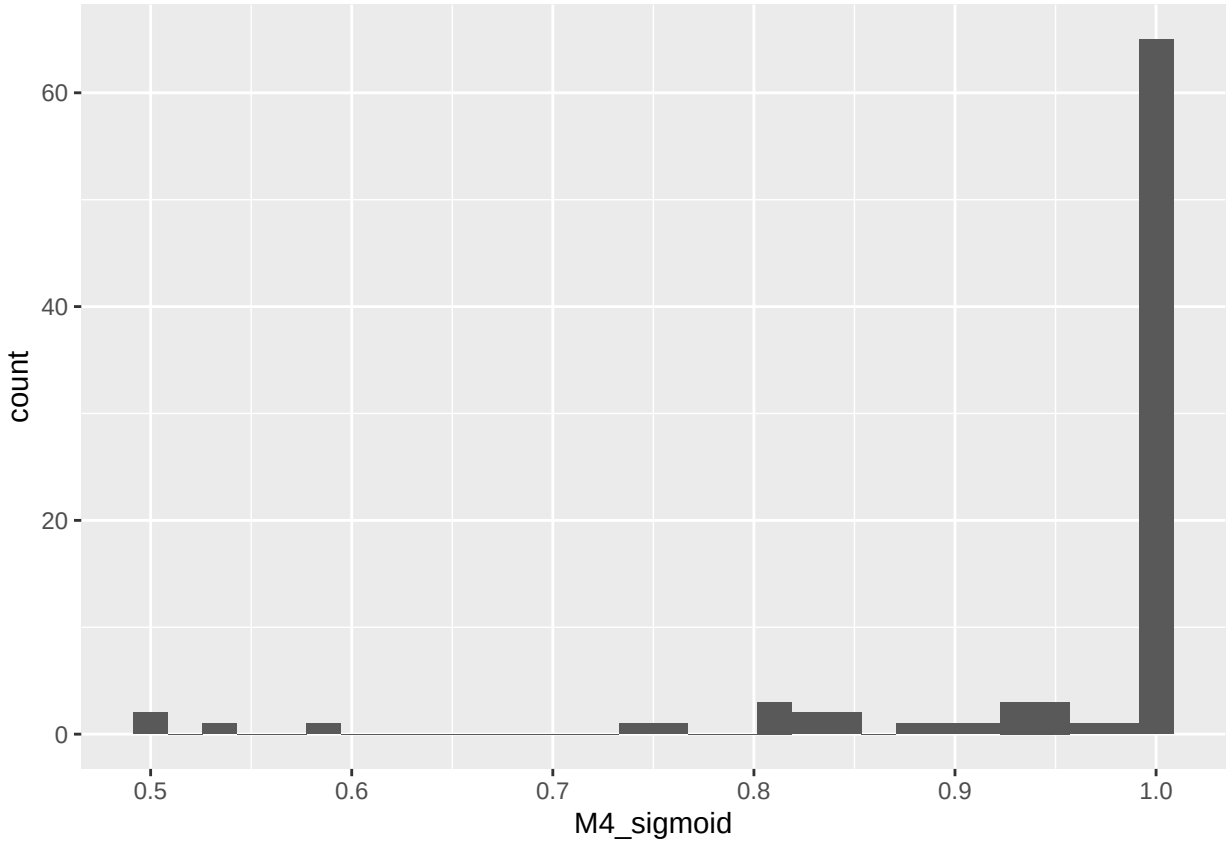

```
ggplot(data = train1, mapping = aes(x = M4)) + geom_histogram()
```

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

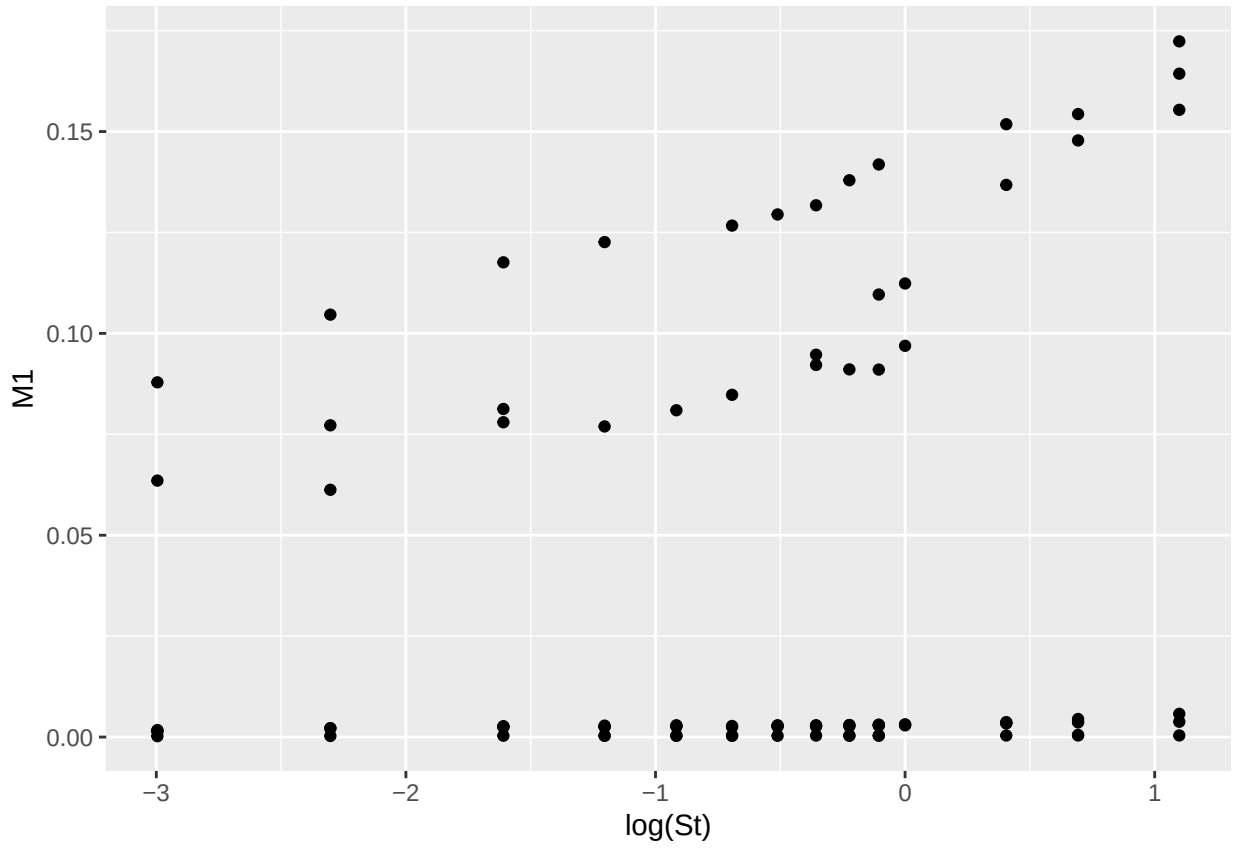



```
ggplot(data = train1, mapping = aes(x = M4_sigmoid)) + geom_histogram()
```

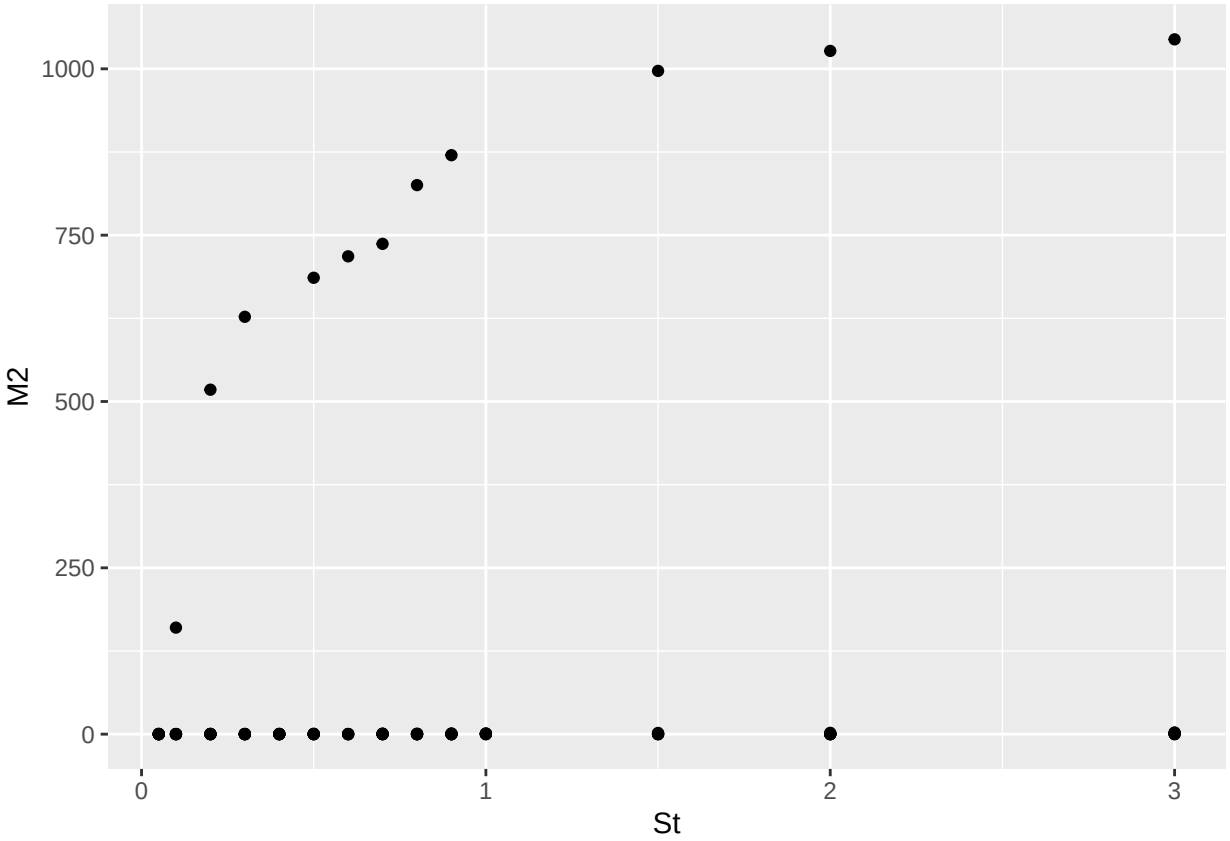
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



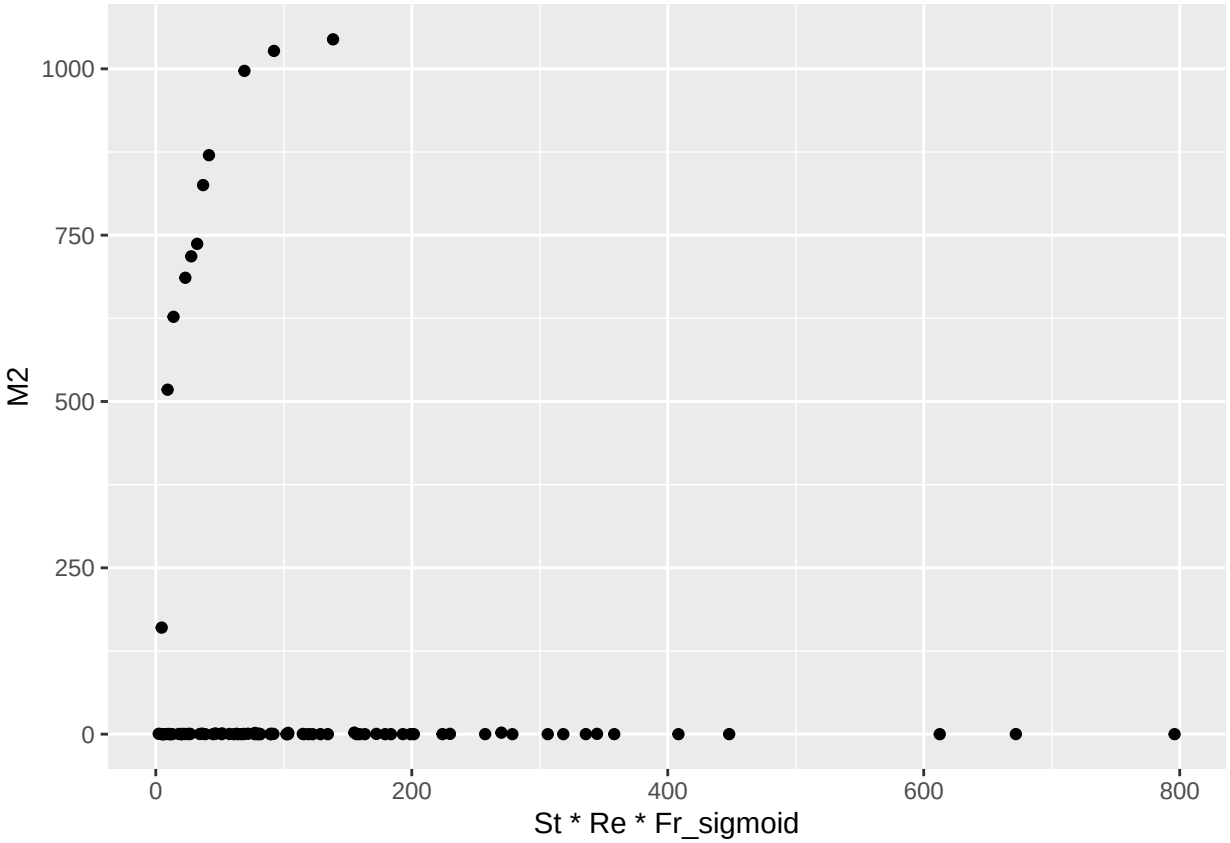

```
ggplot(data = train1, mapping = aes(x = log(St), y = M1)) + geom_point()
```




```
train1 <- train1 %>%
  mutate(Re_categorical = case_when(Re == 90 ~ "Low", Re == 224 ~ "Medium", Re == 398 ~ "High"),
         Fr_categorical = case_when(Fr == 0.052 ~ "Low", Fr == 0.3 ~ "Medium", Fr == Inf ~ "High"))
ggplot(data = train1, mapping = aes(x = St, y = M2)) + geom_point()
```




```
ggplot(data = train1, mapping = aes(x = St*Re*Fr_sigmoid, y = M2)) + geom_point()
```



We will try to create these 4 models:

- **Response**: M1 & **Predictors (Main Effects)**: St, Re, Fr_sigmoid

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

- **Response**: M2 & **Predictors (Main Effects)**: St, Re, Fr_sigmoid, M1

We will attempt to use a combination of subset selection, polynomial, transformation, and interaction variables. We will also include M1 since it is has significant positive relationship with M2 (~0.63).

- **Response**: M3 & **Predictors (Main Effects)**: M2

We know that M2 is almost perfectly correlated (>0.99) with M3, so only using one predictor variable is enough. We try to avoid overfitting by using only M2 as our only predictor to predict M3. We will attempt to use polynomial and transformation variables.

- **Response**: M4 & **Predictors (Main Effects)**: M2, M3

Same reasoning - M2 and M3 are almost perfectly correlated with M4. We will only use these 2 predictors and will attempt to use both transformation and interaction variables (since M2 and M3 are also highly correlated to each other).

Predictive models

```
# Model 1a
model_1a <- glm(M1 ~ St + Re_categorical + Fr_categorical, data = train1)
# summary(model_1a)
with(summary(model_1a), 1 - deviance / null.deviance) # R^2
```

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

```
# 10-fold Cross Validation
set.seed(100)
cv_error_10_1a = rep(0, 10)
for (i in 1:10) {
  model_1a <- glm(M1 ~ St + Re_categorical + Fr_categorical, data = train1)
  cv_error_10_1a[i] = cv.glm(train1, model_1a, K = 10)$delta[1]
  }
sum(cv_error_10_1a) / 10 # MSE
```

```
## [1] 0.0002563688
```

```
# Model 1b
model_1b <- glm(M1 ~ St + Re_categorical + Fr_categorical +
                St * Re_categorical + St * Fr_categorical + Re_categorical * Fr_categorical,
              data = train1)
# summary(model1b)
with(summary(model_1b), 1 - deviance / null.deviance) # R^2
```

```
## [1] 0.9891727
```

```
# 10-fold Cross Validation
set.seed(100)
cv_error_10_1b = rep(0, 10)
for (i in 1:10) {
  model_1b <- glm(M1 ~ St + Re_categorical + Fr_categorical +
                St * Re_categorical + St * Fr_categorical + Re_categorical * Fr_categorical,
              data = train1)
  cv_error_10_1b[i] = cv.glm(train1, model_1b, K = 10)$delta[1]
  }
sum(cv_error_10_1b) / 10 # MSE
```

```
## [1] 6.97473e-05
```

```
# Model 1c
model_1c <- glm(M1_sigmoid ~ St + Re_categorical + Fr_categorical, data = train1)
# summary(model1c)
with(summary(model_1c), 1 - deviance / null.deviance) # R^2
```

```
## [1] 0.929574
```

```r
# 10-fold Cross Validation
set.seed(100)
cv_error_10_1c = rep(0, 10)
for (i in 1:10) {
  model_1c <- glm(M1 ~ St + Re_categorical + Fr_categorical +
                St * Re_categorical + St * Fr_categorical + Re_categorical * Fr_categorical,
              data = train1)
  cv_error_10_1c[i] = cv.glm(train1, model_1c, K = 10)$delta[1]
  }
sum(cv_error_10_1c) / 10 # MSE
```

```
## [1] 6.97473e-05
```

```r
# Model 1d
model_1d <- glm(M1_sigmoid ~ St + Re_categorical + Fr_categorical +
                St * Re_categorical + St * Fr_categorical + Re_categorical * Fr_categorical,
                data = train1)
# summary(model1d)
with(summary(model_1d), 1 - deviance / null.deviance) # R^2
```

```
## [1] 0.9891978
```

```r
# 10-fold Cross Validation
set.seed(100)
cv_error_10_1d = rep(0, 10)
for (i in 1:10) {
  model_1d <- glm(M1 ~ St + Re_categorical + Fr_categorical +
                St * Re_categorical + St * Fr_categorical + Re_categorical * Fr_categorical,
              data = train1)
  cv_error_10_1d[i] = cv.glm(train1, model_1d, K = 10)$delta[1]
  }
sum(cv_error_10_1d) / 10 # MSE
```

```
## [1] 6.97473e-05
```

```r
# Model 2 (linear)
model2 <- lm(M2_sigmoid ~ St + Re_categorical + Fr_categorical + St * Re_categorical + St * Fr_categori
summary(model2)
```

```
##
## Call:
## lm(formula = M2_sigmoid ~ St + Re_categorical + Fr_categorical +
##     St * Re_categorical + St * Fr_categorical + Re_categorical *
##     Fr_categorical, data = train1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.277177 -0.009463  0.001707  0.013402  0.092191
##
## Coefficients: (1 not defined because of singularities)
##                                          Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                                0.489788   0.020407  24.001  < 2e-16
## St                                         0.018040   0.019090   0.945 0.347639
## Re_categoricalLow                          0.071319   0.026622   2.679 0.009049
## Re_categoricalMedium                       0.002200   0.025947   0.085 0.932646
## Fr_categoricalLow                          0.017554   0.025750   0.682 0.497492
## Fr_categoricalMedium                      -0.015098   0.024765  -0.610 0.543907
## St:Re_categoricalLow                       0.079519   0.019480   4.082 0.000109
## St:Re_categoricalMedium                    0.003769   0.019255   0.196 0.845335
## St:Fr_categoricalLow                      -0.024122   0.015509  -1.555 0.124023
## St:Fr_categoricalMedium                    0.024649   0.019123   1.289 0.201308
## Re_categoricalLow:Fr_categoricalLow        0.325373   0.031491  10.332 3.91e-16
## Re_categoricalMedium:Fr_categoricalLow     0.067516   0.030039   2.248 0.027503
## Re_categoricalLow:Fr_categoricalMedium     0.039699   0.030094   1.319 0.191077
## Re_categoricalMedium:Fr_categoricalMedium        NA         NA      NA       NA
##
## (Intercept)                               ***
## St
## Re_categoricalLow                         **
## Re_categoricalMedium
## Fr_categoricalLow
## Fr_categoricalMedium
## St:Re_categoricalLow                      ***
## St:Re_categoricalMedium
## St:Fr_categoricalLow
## St:Fr_categoricalMedium
## Re_categoricalLow:Fr_categoricalLow       ***
## Re_categoricalMedium:Fr_categoricalLow    *
## Re_categoricalLow:Fr_categoricalMedium
## Re_categoricalMedium:Fr_categoricalMedium
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0486 on 76 degrees of freedom
## Multiple R-squared:  0.9297, Adjusted R-squared:  0.9186
## F-statistic: 83.74 on 12 and 76 DF,  p-value: < 2.2e-16
```

```r
# vif(model2)

# Model 2 interactions
model2_int1 <- lm(M2_sigmoid ~ St + Re + Fr_sigmoid + St*Re + St*Fr_sigmoid + Re*Fr_sigmoid, data = tra
summary(model2_int1)
```

```
##
## Call:
## lm(formula = M2_sigmoid ~ St + Re + Fr_sigmoid + St * Re + St *
##     Fr_sigmoid + Re * Fr_sigmoid, data = train1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20051 -0.07816 -0.01241  0.05618  0.25589
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.1303328  0.0963556  11.731  < 2e-16 ***
```

```
## St             0.1173838  0.0541272   2.169    0.033 *
## Re            -0.0019675  0.0003647  -5.395 6.48e-07 ***
## Fr_sigmoid    -0.5873023  0.1269051  -4.628 1.37e-05 ***
## St:Re         -0.0002900  0.0001279  -2.267    0.026 *
## St:Fr_sigmoid -0.0135248  0.0652887  -0.207    0.836
## Re:Fr_sigmoid  0.0018129  0.0004418   4.104 9.55e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1056 on 82 degrees of freedom
## Multiple R-squared:  0.6419, Adjusted R-squared:  0.6157
## F-statistic:  24.5 on 6 and 82 DF,  p-value: < 2.2e-16
```

```
vif(model2_int1)
```

```
##            St            Re    Fr_sigmoid         St:Re St:Fr_sigmoid
##     14.289023     13.402390      6.235856      5.394906     11.596560
## Re:Fr_sigmoid
##     16.278692
```

Apply to test data