Case Study Report

02 November, 2022

```
data_train<-read_csv("data/data-train.csv")
data_test<-read_csv("data/data-test.csv")
```

Introduction

The objective of this case study is to develop a predictive model to predict the distributions of particle clusters in turbulence from three predictors: Reynold's number Re, gravitational acceleration Fr, and particle characteristics (size or density which is quantified by Stoke's number St).

Developing an understanding of turbulence is important because the effects of turbulence are present in a wide variety of problems. For example, the distribution of ash in the atmosphere is controlled by atmospheric turbulence, which has many implications for the environment and aviation. Turbulence also controls the population dynamics of planktons, which play an important role in the carbon cycle. On a more cosmological and atmospheric level, turbulence plays a central part in the dispersion of ash in the atmosphere which has many implications for the environment and aviation as well as the thermodynamics of clouds, radiative properties, and the rate at which droplets grow to form rain. The model we have decided to use to predict turbulence is a linear regression with interaction terms, shown below:

$St + Re_{category} + Fr_{transformed} + Fr_{transformed} * Re_{category} + St * Re_{category}$

where $Re_{category}$ is a variable that classifies the training Re into three categories ($Re = 90$: Low, $Re = 224$: Medium", $Re = 398$: High)

EDAs justifying the model

Methodology

EDA

After loading the data, we performed exploratory data analysis on all three predictors and four moments.

We first noted that the predictor variables `Re` is clustered at fixed values, with `Re` clustering at 90, 224 and 398 (3 levels).

```
summary(data_train$St)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0500  0.3000  0.7000  0.8596  1.0000  3.0000
```

```
summary(data_train$Re)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    90.0    90.0   224.0   214.5   224.0   398.0
```

```
summary(data_train$Fr)
```

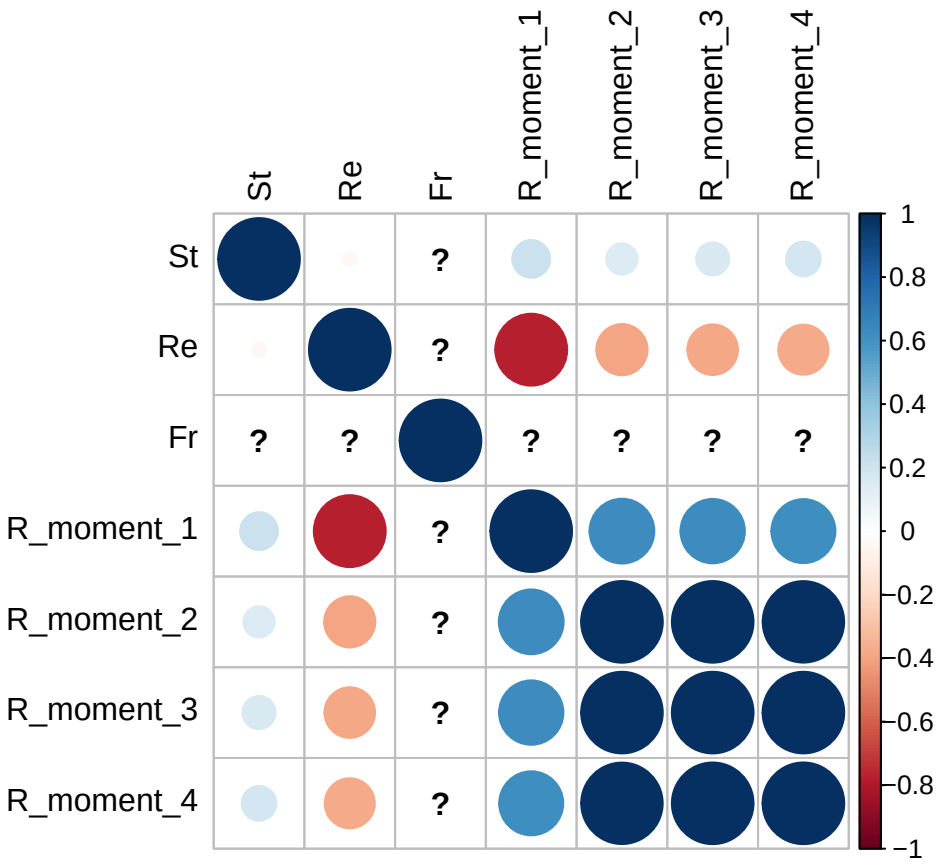
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.052   0.052   0.300     Inf     Inf     Inf
```

We found that the moments are not only highly correlated,

```
res<-cor(data_train)
res
```

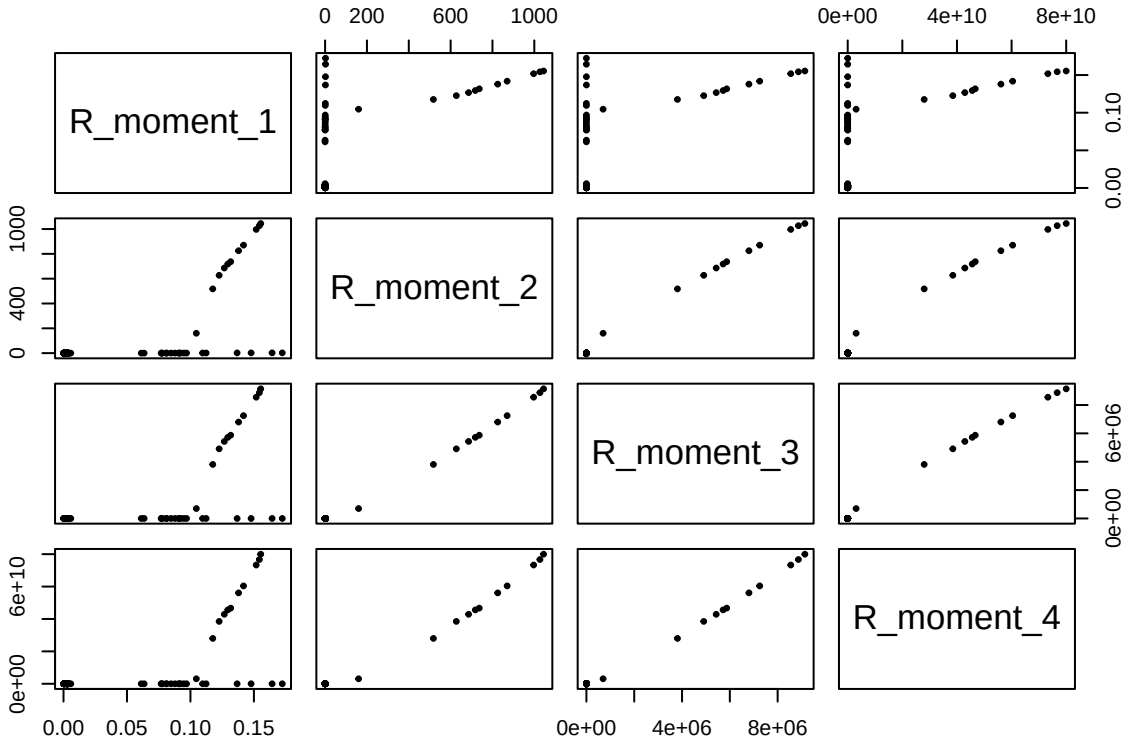
```
##                      St          Re  Fr R_moment_1 R_moment_2 R_moment_3
## St           1.00000000 -0.03169871 NaN  0.2147681  0.1479257  0.1647465
## Re          -0.03169871  1.00000000 NaN -0.7747206 -0.3932344 -0.3844289
## Fr                  NaN         NaN   1        NaN        NaN        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
```

```
corrplot(res, tl.col ="black")
```



but that they are linearly correlated:

```
pairs(data_train[4:7], cex = 0.5, pch = 19)
```



Therefore, we decided to fit a model on `R_moment_1`, which will give us the relationship of the predictor variables on the other moments due to the high linear correlation between the moments.

Moreover, we noticed that the gravitational acceleration has infinite values, which is problematic. Therefore, we used inverse logit transform on `Fr` to transform the infinity value into a finite value (Inf transformed to 1):

```
data_train <- data_train %>%
  mutate(Re_category = case_when(Re == 90 ~ "Low", Re==224 ~ "Medium", Re == 398 ~ "High"))%>%
  mutate(Fr_transformed = invlogit(Fr))
```

We then explored shrinkage methods such as ridge regression and lasso. However, since we know that the three predictors are all active so that we do not need predictor selection, so we attempted to fit a ridge regression model:

Ridge Model

```
y <- data_train$R_moment_1
x <- data.matrix(data_train[, c('Re', 'St', 'Fr_transformed')])
```

```
set.seed(123)
sample <- sample(c(TRUE, FALSE), nrow(data_train), replace=TRUE)
train <- sample(1:nrow(x), nrow(x)/2)
test <- (-train)
y.test <- y[test]
```

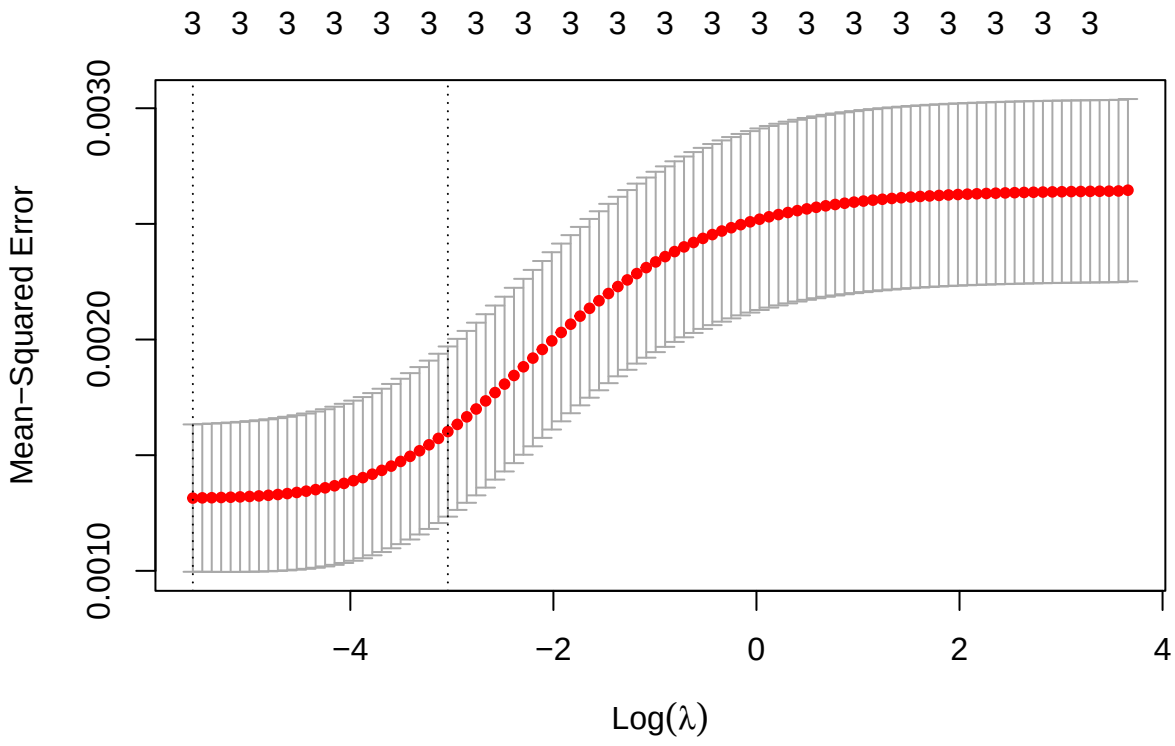
```
lambda_seq = 10^seq(10, -2, length = 100)

ridgemodel <- glmnet(x[train,], y[train], alpha = 0, lambda = lambda_seq)

summary(ridgemodel)
```

```
##           Length Class     Mode
## a0        100    -none-    numeric
## beta      300    dgCMatrix S4
## df        100    -none-    numeric
## dim         2    -none-    numeric
## lambda    100    -none-    numeric
## dev.ratio 100    -none-    numeric
## nulldev     1    -none-    numeric
## npasses     1    -none-    numeric
## jerr        1    -none-    numeric
## offset      1    -none-    logical
## call        5    -none-    call
## nobs        1    -none-    numeric
```

```
set.seed(123)
#perform k-fold cross-validation to find optimal lambda value
cv_ridgemodel <- cv.glmnet(x[train,], y[train], alpha = 0)
plot(cv_ridgemodel)
```



```
set.seed(123)
#find optimal lambda value that minimizes test MSE
best_lambda <- cv_ridgemodel$lambda.min
best_lambda
```

```
## [1] 0.003882444
```

```
ridge.pred <- predict(ridgemodel, s = best_lambda, newx = x[test,])
mse <- mean((ridge.pred - y.test)^2)
mse
```

```
## [1] 0.001498519
```

The ridge regression we fitted through cross-validation gives us an MSE of 0.001498519

```
lm.fit_M1 <- lm(R_moment_1 ~ St+ Re_category+Fr_transformed+Fr_transformed*Re_category+St*Re_category, d
lm_summary_M1<-summary(lm.fit_M1)
lm_mse <-mean(lm_summary_M1$residual^2)
final_model_M1<-lm.fit_M1
lm_mse
```

```
## [1] 7.656106e-05
```

By fitting a simple linear regression and adding interaction terms, we obtained a model that has a MSE of 7.656106e-05, which is much smaller than the ridge regression MSE. Moreover, the model fits the data closely with R^2 value of 0.9727. Therefore, we decided to use this model as our final model:

```
library(jtools) # Load jtools
summ(final_model_M1)
```

```
## MODEL INFO:
## Observations: 89
## Dependent Variable: R_moment_1
## Type: OLS linear regression
##
## MODEL FIT:
## F(8,80) = 392.74, p = 0.00
## R² = 0.98
## Adj. R² = 0.97
##
## Standard errors: OLS
## ------------------------------------------------------------------------
##                                        Est.   S.E.   t val.      p
## ------------------------------------- ------- ------ -------- ------
## (Intercept)                            0.00   0.01     0.03   0.98
## St                                     0.00   0.00     0.02   0.98
## Re_categoryLow                         0.12   0.01    12.27   0.00
## Re_categoryMedium                      0.00   0.01     0.16   0.87
## Fr_transformed                         0.00   0.01     0.01   0.99
## Re_categoryLow:Fr_transformed         -0.05   0.01    -4.39   0.00
## Re_categoryMedium:Fr_transformed       0.00   0.01     0.05   0.96
```

```
## St:Re_categoryLow                          0.03   0.00     7.99   0.00
## St:Re_categoryMedium                       0.00   0.00     0.23   0.82
## ----------------------------------------------------------------------
```

We fitted models on the same model to predict second, third and fourth moments due to the collinearity as explained above:

```
data_train <- data_train %>%
  mutate(Fr_category = case_when(Fr == 0.052 ~ "Low", Fr == 0.3 ~ "Medium", Fr == Inf ~ "High"))

final_model_M2 <- lm(R_moment_2 ~ St+ Re_category+Fr_category+Fr_category*Re_category+St*Re_category, da
final_model_M3 <- lm(R_moment_2 ~ St+ Re_category+Fr_category+Fr_category*Re_category+St*Re_category, da
final_model_M4 <- lm(R_moment_2 ~ St+ Re_category+Fr_category+Fr_category*Re_category+St*Re_category, da
summ(final_model_M2)
```

```
## MODEL INFO:
## Observations: 89
## Dependent Variable: R_moment_2
## Type: OLS linear regression
##
## MODEL FIT:
## F(10,78) = 43.93, p = 0.00
## R² = 0.85
## Adj. R² = 0.83
##
## Standard errors: OLS
## ------------------------------------------------------------------------------
##                                            Est.    S.E.   t val.      p
## ----------------------------------------- -------- ------- -------- ------
## (Intercept)                                  0.01   42.26     0.00   1.00
## St                                           0.00   34.07     0.00   1.00
## Re_categoryLow                             -84.75   58.31    -1.45   0.15
## Re_categoryMedium                           -0.02   57.24    -0.00   1.00
## Fr_categoryLow                              -0.00   51.11    -0.00   1.00
## Fr_categoryMedium                            0.00   43.67     0.00   1.00
## Re_categoryLow:Fr_categoryLow              672.21   69.90     9.62   0.00
## Re_categoryMedium:Fr_categoryLow             0.26   66.50     0.00   1.00
## Re_categoryLow:Fr_categoryMedium           -26.49   66.33    -0.40   0.69
## Re_categoryMedium:Fr_categoryMedium
## St:Re_categoryLow                          108.99   40.99     2.66   0.01
## St:Re_categoryMedium                         0.06   41.60     0.00   1.00
## ------------------------------------------------------------------------------
```

```
summ(final_model_M3)
```

```
## MODEL INFO:
## Observations: 89
## Dependent Variable: R_moment_2
## Type: OLS linear regression
##
## MODEL FIT:
## F(10,78) = 43.93, p = 0.00
## R² = 0.85
```

```
## Adj. R² = 0.83
##
## Standard errors: OLS
## ------------------------------------------------------------------------------
##                                              Est.    S.E.   t val.      p
## ---------------------------------------- -------- ------- -------- ------
## (Intercept)                                  0.01   42.26     0.00   1.00
## St                                           0.00   34.07     0.00   1.00
## Re_categoryLow                             -84.75   58.31    -1.45   0.15
## Re_categoryMedium                           -0.02   57.24    -0.00   1.00
## Fr_categoryLow                              -0.00   51.11    -0.00   1.00
## Fr_categoryMedium                            0.00   43.67     0.00   1.00
## Re_categoryLow:Fr_categoryLow              672.21   69.90     9.62   0.00
## Re_categoryMedium:Fr_categoryLow             0.26   66.50     0.00   1.00
## Re_categoryLow:Fr_categoryMedium           -26.49   66.33    -0.40   0.69
## Re_categoryMedium:Fr_categoryMedium
## St:Re_categoryLow                          108.99   40.99     2.66   0.01
## St:Re_categoryMedium                         0.06   41.60     0.00   1.00
## ------------------------------------------------------------------------------
```

```
summ(final_model_M4)
```

```
## MODEL INFO:
## Observations: 89
## Dependent Variable: R_moment_2
## Type: OLS linear regression
##
## MODEL FIT:
## F(10,78) = 43.93, p = 0.00
## R² = 0.85
## Adj. R² = 0.83
##
## Standard errors: OLS
## ------------------------------------------------------------------------------
##                                              Est.    S.E.   t val.      p
## ---------------------------------------- -------- ------- -------- ------
## (Intercept)                                  0.01   42.26     0.00   1.00
## St                                           0.00   34.07     0.00   1.00
## Re_categoryLow                             -84.75   58.31    -1.45   0.15
## Re_categoryMedium                           -0.02   57.24    -0.00   1.00
## Fr_categoryLow                              -0.00   51.11    -0.00   1.00
## Fr_categoryMedium                            0.00   43.67     0.00   1.00
## Re_categoryLow:Fr_categoryLow              672.21   69.90     9.62   0.00
## Re_categoryMedium:Fr_categoryLow             0.26   66.50     0.00   1.00
## Re_categoryLow:Fr_categoryMedium           -26.49   66.33    -0.40   0.69
## Re_categoryMedium:Fr_categoryMedium
## St:Re_categoryLow                          108.99   40.99     2.66   0.01
## St:Re_categoryMedium                         0.06   41.60     0.00   1.00
## ------------------------------------------------------------------------------
```

Here we perform a 5-fold cross validation on the model.

```r
train.control <- trainControl(method = "cv", number = 5)

M1_cv <- train(R_moment_1 ~ St+ Re_category+Fr_transformed+Fr_transformed*Re_category+St*Re_category, da
print(M1_cv)
```

```
## Linear Regression
##
## 89 samples
##  3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 73, 72, 69, 72, 70
## Resampling results:
##
##   RMSE         Rsquared   MAE
##   0.009227751  0.9768461  0.004715023
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
M2_cv <- train(R_moment_2 ~ St+Re_category+Fr_category+Fr_category*Re_category+St*Re_category, data = da
M3_cv <- train(R_moment_3 ~ St+Re_category+Fr_category+Fr_category*Re_category+St*Re_category, data = da
M4_cv <- train(R_moment_4 ~ St+Re_category+Fr_category+Fr_category*Re_category+St*Re_category, data = da
print(M2_cv)
```

```
## Linear Regression
##
## 89 samples
##  3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 72, 70, 71, 71, 72
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   109.2029  0.8054879  46.19636
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
print(M3_cv)
```

```
## Linear Regression
##
## 89 samples
##  3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 72, 71, 71, 71, 71
## Resampling results:
```

```
##
##   RMSE     Rsquared   MAE
##   1127018  0.7371293  487520.1
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
print(M4_cv)
```

```
## Linear Regression
##
## 89 samples
##  3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 70, 72, 71, 71, 72
## Resampling results:
##
##   RMSE        Rsquared   MAE
##   8619535832  0.7804712  4118026380
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
library(knitr)
M1_metrics <- M1_cv$results %>% select(c("RMSE", "Rsquared", "MAE"))
M2_metrics <- M2_cv$results %>% select(c("RMSE", "Rsquared", "MAE"))
M3_metrics <- M3_cv$results %>% select(c("RMSE", "Rsquared", "MAE"))
M4_metrics <- M4_cv$results %>% select(c("RMSE", "Rsquared", "MAE"))

cv_df <- merge(M1_metrics, M2_metrics, by=c("RMSE", "Rsquared", "MAE"), all = TRUE)
cv_df2 <- merge(cv_df, M3_metrics, by=c("RMSE", "Rsquared", "MAE"), all = TRUE)
total_cv <- merge(cv_df2, M4_metrics, by=c("RMSE", "Rsquared", "MAE"), all = TRUE)
total_cv <- total_cv %>% mutate(Model = c('M1', 'M2', 'M3', 'M4')) %>% relocate(Model, .before = RMSE)
total_cv %>% kable(digits = 3)
```

Model	RMSE	Rsquared	MAE
M1	9.000000e-03	0.977	5.000000e-03
M2	1.092030e+02	0.805	4.619600e+01
M3	1.127018e+06	0.737	4.875201e+05
M4	8.619536e+09	0.780	4.118026e+09

Results

We made predictions on the hold-out set in data-test.csv, and generated a csv file containing the predictions for the first, second, third and fourth moments.

```
data_test_transformed <- data_test %>%
  mutate(Re_category = case_when(Re == 90 ~ "Low", Re==224 ~ "Medium", Re == 398 ~ "High"))%>%
  mutate(Fr_transformed = invlogit(Fr))%>%
  mutate(Fr_category = case_when(Fr == 0.052 ~ "Low", Fr == 0.3 ~ "Medium", Fr == Inf ~ "High"))%>%
```

```
  mutate(Predicted_M1 = predict(final_model_M1, .))%>%
  mutate(Predicted_M2 = predict(final_model_M2, .))%>%
  mutate(Predicted_M3 = predict(final_model_M3, .))%>%
  mutate(Predicted_M4 = predict(final_model_M4, .))

data_predicted_output <- data_test_transformed%>%select(St, Re, Fr, Predicted_M1, Predicted_M2, Predicte
write.csv(data_predicted_output,"data-predict.csv", row.names = FALSE)
data_predicted_output %>% kable(digits = 5)
```

St	Re	Fr	Predicted_M1	Predicted_M2	Predicted_M3	Predicted_M4
0.05	398	0.052	0.00027	0.00345	0.00345	0.00345
0.20	398	0.052	0.00028	0.00366	0.00366	0.00366
0.70	398	0.052	0.00031	0.00436	0.00436	0.00436
1.00	398	0.052	0.00033	0.00479	0.00479	0.00479
0.10	398	Inf	0.00033	0.00528	0.00528	0.00528
0.60	398	Inf	0.00037	0.00599	0.00599	0.00599
1.00	398	Inf	0.00039	0.00655	0.00655	0.00655
1.50	398	Inf	0.00043	0.00725	0.00725	0.00725
3.00	398	Inf	0.00053	0.00937	0.00937	0.00937
3.00	224	0.300	0.00476	0.18816	0.18816	0.18816
0.10	224	Inf	0.00247	-0.00485	-0.00485	-0.00485
0.50	224	Inf	0.00282	0.02140	0.02140	0.02140
0.40	90	0.052	0.10642	631.05620	631.05620	631.05620
1.00	90	0.052	0.12325	696.45211	696.45211	696.45211
0.05	90	0.300	0.09339	-105.78907	-105.78907	-105.78907
0.30	90	0.300	0.10040	-78.54077	-78.54077	-78.54077
0.60	90	0.300	0.10882	-45.84282	-45.84282	-45.84282
0.80	90	0.300	0.11442	-24.04418	-24.04418	-24.04418
0.40	90	Inf	0.08092	-41.15218	-41.15218	-41.15218
0.50	90	Inf	0.08373	-30.25286	-30.25286	-30.25286
0.60	90	Inf	0.08653	-19.35354	-19.35354	-19.35354
1.50	90	Inf	0.11177	78.74032	78.74032	78.74032
2.00	90	Inf	0.12579	133.23691	133.23691	133.23691

Major observations from the results:

-

results section discussing your predictive results (don't forget uncertainty!), as well as insights on the scientific problem.

Conclusion