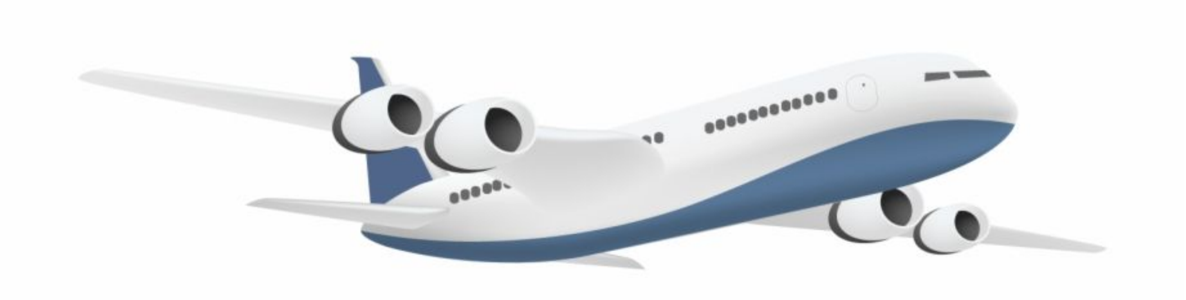


DATA EXPEDITION LAB

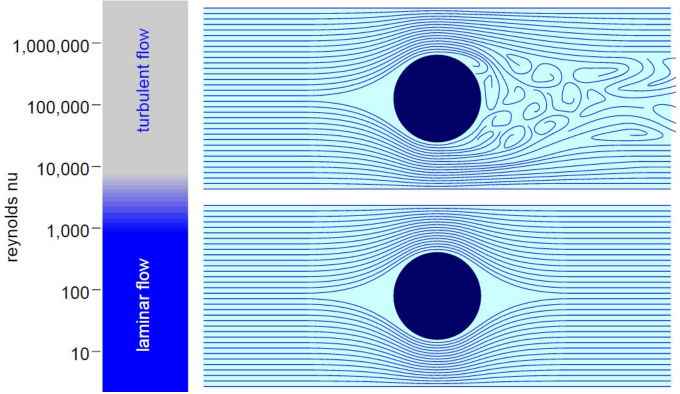
Mihir Dutta, Michael Nicholson, Judin Thomas

Problem

Modeling Particle Clustering

Turbulence: Referred to by Feynman as the most important unsolved problem in physics

- Turbulence can be observed in everyday life such as the surf break on a beach, the rapids in a rocky river, and the distribution of creamer in one's coffee.
- Turbulence is very computationally expensive to model so our goal is to apply machine learning methods in order to generate an efficient way to predict and interpret particle clustering in a closed system



Variables and Distribution

Explanatory Variables:

- Reynolds Number: measure of turbulence intensity
- Froud Number: measure of gravitational acceleration
- Stokes Number: measure of the size/density of particles

Response Variables:

- Because of computational constraints, instead of modeling the entire probability distribution of particle clusters we will attempt to model the first, second, third, and fourth moments

Methodology

To create our models, we used two different methodologies:

- **Generalized Additive Model:**
  - Began with simple regression model and added complexity by adding polynomial degrees and interactions in a stepwise fashion, ultimately using ANOVA for model selection.
- **Random Forest:**
  - Created four random forest models to predict each of the four moments and analyzed feature importance.

General Additive Model (GAM)

Why GAM?

Interpretability

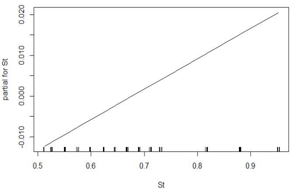
- The additive nature of this model allows us to interpret the impact of each input separately from the other variables, which is especially valuable for scientific inference
- The relationship of each input on the output can be seen with the plot function in R

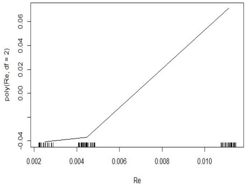
Analyzing Interaction Effects

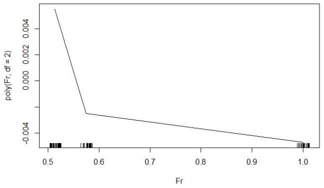
- Interactions can be added to the model to better understand how the different inputs affect each other
- However, adding interactions reduces interpretability

Flexibility

- Using a GAM allows us to use a number of fits for each input, such as polynomials, splines, etc.







Model Output

```
Call:
lm(formula = R_moment_1 ~ St + poly(Re, df = 2) + poly(Fr, df = 2) +
    Re:Fr + St:Re, data = data)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0211684 -0.0046996 -0.0004214  0.0053228  0.0265414

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       0.024728   0.006806   3.633 0.000489 ***
St               -0.086685   0.014324  -6.052 4.24e-08 ***
poly(Re, df = 2)1  0.097622   0.053021   1.841 0.069257 .
poly(Re, df = 2)2  0.080872   0.008491   9.525 7.31e-15 ***
poly(Fr, df = 2)1  0.051076   0.016412   3.112 0.002566 **
poly(Fr, df = 2)2  0.042024   0.008562   4.908 4.69e-06 ***
Re:Fr            -6.684729   1.123369  -5.951 6.53e-08 ***
St:Re            23.940957   1.885719  12.696  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.008129 on 81 degrees of freedom
Multiple R-squared:  0.9805,    Adjusted R-squared:  0.9788
F-statistic: 581.7 on 7 and 81 DF,  p-value: < 2.2e-16
```

GAM Models

gam(R_moment_1 ~ St + poly(Re, df=2) + poly(Fr, df = 2) + Re:Fr + St:Re)

gam(R_moment_2 ~ St + Re + poly(Fr, df = 2) + Re:Fr + St:Re)

gam(R_moment_3 ~ St + Re + poly(Fr, df = 2) + Re:Fr + St:Re + St:Fr)

gam(R_moment_4 ~ St + Re + poly(Fr, df = 2) + Re:Fr + St:Re + St:Fr)

General Additive Model (GAM)

ANOVA

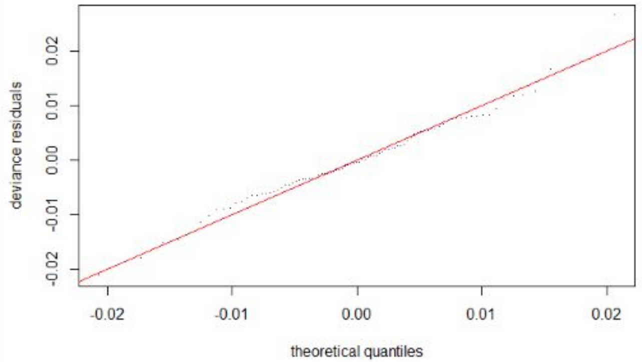
Additive Model

```
Analysis of Variance Table

Model 1: R_moment_1 ~ St + Re + Fr
Model 2: R_moment_1 ~ St + Re + poly(Fr, df = 2)
Model 3: R_moment_1 ~ St + poly(Re, df = 2) + poly(Fr, df = 2)
Model 4: R_moment_1 ~ poly(St, df = 2) + poly(Re, df = 2) + poly(Fr, df = 2)
Model 5: R_moment_1 ~ ns(St, df = 3) + poly(Re, df = 2) + poly(Fr, df = 2)
  Res.Df      RSS Df Sum of Sq       F    Pr(>F)
1     85 0.029499
2     84 0.027155  1 0.0023442  9.9299  0.002279 **
3     83 0.019550  1 0.0076052 32.2149 2.077e-07 ***
4     82 0.019501  1 0.0000482  0.2041  0.652642
5     81 0.019122  1 0.0003791  1.6058  0.208717
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interactions

```
Analysis of Variance Table

Model 1: R_moment_1 ~ St + poly(Re, df = 2) + poly(Fr, df = 2)
Model 2: R_moment_1 ~ St + poly(Re, df = 2) + poly(Fr, df = 2) + Re:Fr
Model 3: R_moment_1 ~ St + poly(Re, df = 2) + poly(Fr, df = 2) + Re:Fr +
    St:Re
Model 4: R_moment_1 ~ St + poly(Re, df = 2) + poly(Fr, df = 2) + Re:Fr +
    St:Re + St:Fr
  Res.Df       RSS Df Sum of Sq        F    Pr(>F)
1     83 0.0195496
2     82 0.0160047  1 0.0035449  54.1269 1.451e-10 ***
3     81 0.0053528  1 0.0106519 162.6415 < 2.2e-16 ***
4     80 0.0052394  1 0.0001134   1.7311     0.192
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual Plot



Choice of Model

- Fitted gam models with increasing degrees for each input
  - Used ANOVA to select the best model of those fitted
- Iteratively added interactions to the model
  - Interactions reduce the interpretability of the model but can improve predictive accuracy
  - We decided the loss of interpretability was outweighed by the increase of predictive accuracy

Random Forest

Why Random Forest?

Predictive Power

- The predictive performance can compete with the best supervised learning algorithms

Analyzing Feature Importance

- Random Forests are a good model to analyze feature importance
- Uses out-of-bag error to measure change in MSE by permuting inputs

Test Error Estimation

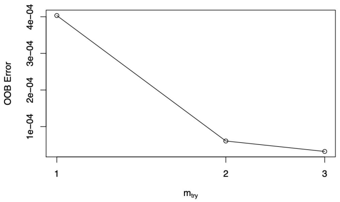
- Computationally efficient way of analyzing test error
- Random forests are repeatedly tested on data not trained on to estimate test error

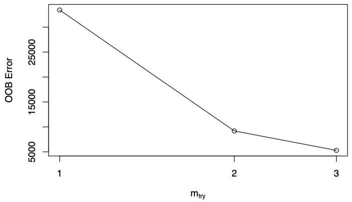
Models

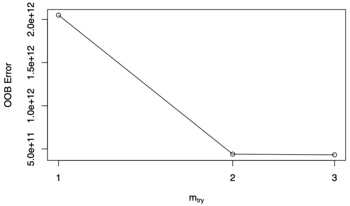
Random Forest Models

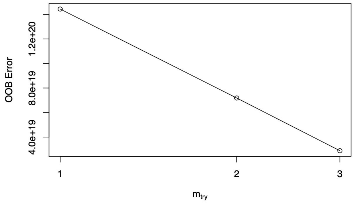
- R1_model <- randomForest(R_moment_1 ~ St + Re + Fr, data=train, mtry=3, importance=TRUE, na.action=na.omit)
- R2_model <- randomForest(R_moment_2 ~ St + Re + Fr, data=train, mtry=3, importance=TRUE, na.action=na.omit)
- R3_model <- randomForest(R_moment_3 ~ St + Re + Fr, data=train, mtry=3, importance=TRUE, na.action=na.omit)
- R4_model <- randomForest(R_moment_4 ~ St + Re + Fr, data=train, mtry=3, importance=TRUE, na.action=na.omit)

Tuning









Bagging

Feature Importance

First Moment Model

```
##        %IncMSE
## St 34.28927
## Re 84.47860
## Fr 33.51107
```

Second Moment Model

```
##        %IncMSE
## St 18.93854
## Re 49.66762
## Fr 48.95960
```

Third Moment Model

```
##        %IncMSE
## St 17.23641
## Re 47.38180
## Fr 44.73023
```

Fourth Moment Model

```
##        %IncMSE
## St 20.25196
## Re 44.98411
## Fr 41.86153
```

Predicted Values

	St	Re	Fr	R1_pred	R2_pred	R3_pred	R4_pred
1	0.05	398	0.512997071458542	0.000262583525000001	0.00336655486333385	0.090808455996681	3.33176537322998
2	0.2	398	0.512997071458542	0.000295862801	0.00435870401761747	0.0925165359592065	19.0236613616943
3	0.7	398	0.512997071458542	0.000325200658666666	0.00558801132999668	0.290092357165413	14.2734321346283
4	1	398	0.512997071458542	0.000356907540333332	0.00692517062999607	3625.42962379754	30.3848154220581
5	0.1	398	1	0.000282082229333335	0.00326038627333531	0.101239946731832	7.48766680526733
6	0.6	398	1	0.000353320040333334	0.00616477795333353	0.136176187152741	8.9767612247467
7	1	398	1	0.000372390383333332	0.00680892470333433	0.141961943856906	15.8933089809418
8	1.5	398	1	0.000384052737	0.0115851004500009	0.780762590026949	46.2512295665741
9	3	398	1	0.000390688661666667	1.02112576444	38395.5900116278	360000448.795939
10	3	224	0.574442516811659	0.00432924455333332	1.08402653850666	38396.8488593861	516700573.406896
11	0.1	224	1	0.00210639868333333	0.0413675987666684	0.84380311452481	15.7596069755554
12	0.5	224	1	0.00285469269	0.0441782523666677	0.87356437879568	17.9404273204803
13	0.4	90	0.512997071458542	0.122813541733333	652.73851092	5088104.11405533	39812934678.696
14	1	90	0.512997071458542	0.134121519466667	822.548033860001	6792527.82599158	56294087015.1516
15	0.05	90	0.574442516811659	0.0694705504000001	0.179981255066665	467.564808244654	2456040.02870487
16	0.3	90	0.574442516811659	0.0775555698000001	0.24971714493333	2.23177943866118	54.2412788543701
17	0.6	90	0.574442516811659	0.0907985521666664	0.587337788933329	7.63565836535324	81.4388767642975
18	0.8	90	0.574442516811659	0.0995892607999998	0.705775886266662	9.26300678466796	99.5135005626678
19	0.4	90	1	0.0810682012666667	0.403186284666663	4.58212561134319	61.1097030658722
20	0.5	90	1	0.0846349172666665	0.477836953299997	5.84285649135499	70.0973043403626
21	0.6	90	1	0.0879439161666665	0.54408981493333	6.76410376535752	73.4923890800476
22	1.5	90	1	0.138772824666667	1.61273951126666	17128.617952278	146600502.028502
23	2	90	1	0.1490049134	2.73423135599999	50389.6050595707	692190624.624589

Results From Comparing Models

Comparing Models

Test Set Validation

- To compare the MSE's of the Generalized Additive Model to the Random Forest/Bagging model, we split the the data into two groups, 80% for training and 20% for testing
- We trained on the training set, predicted on the test set, and calculated the MSE for each of the four moments

Moment	Bagging MSE - raw moments	GAM MSE - raw moments	Bagging MSE - log(moments)
1st	**$1.165 * 10^{-5}$**	**$1.131 * 10^{-4}$**	**28.124**
2nd	**3,481**	**42,211**	**81,832**
3rd	**$3.785 * 10^{11}$**	**$3.233 * 10^{12}$**	**$5.412 * 10^{13}$**
4th	**$2.534 * 10^{19}$**	**$2.52 * 10^{20}$**	**$2.55 * 10^{20}$**

MSE Conclusions

It is important to note that the reasons the MSE's are so large is because there was a lot of variance within the moments.

However, we can see that across the board, that the Random Forest/Bagging performed better than the Generalized Additive Model, with a difference in an order of magnitude. It is clear from the MSE values that the Random Forest/Bagging is a better predictive model, so we decided to use this model in our predictions on the true test data set.

Reasons for MSE Disparity

- We believe there could have been an overfitting of GAM due to sparse and highly varied moment values
- The GAM performed much poorer than the models which incorporated interaction effects
- The greater flexibility and complexity of the Forest/Bagging tailors better for this sparse and highly varied data

Overall, we can see that the Bagging predictive model performs much better with this data than the GAM

Conclusions

We decided to use the Bagging model over the GAM model because of its lower hold out test MSE

Conclusions

We believe that the Bagging model has stronger predictive capabilities due to this lower error level

Conclusions

We lose some interpretability by using such a complex model, but we believe the tradeoff for predictive accuracy is worth it. We also can gain some interpretable information from the model with importance of variables.