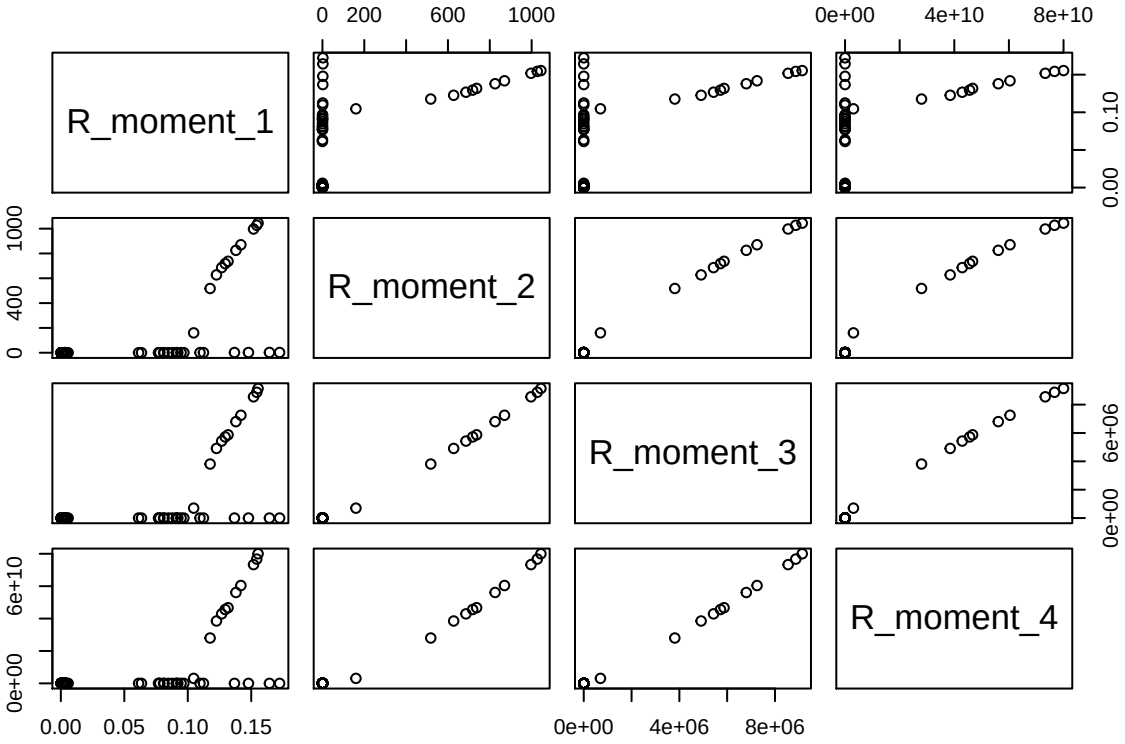
case_study_report

Connie Wu, Jason McEachin, Joe Choo, Scott Heng

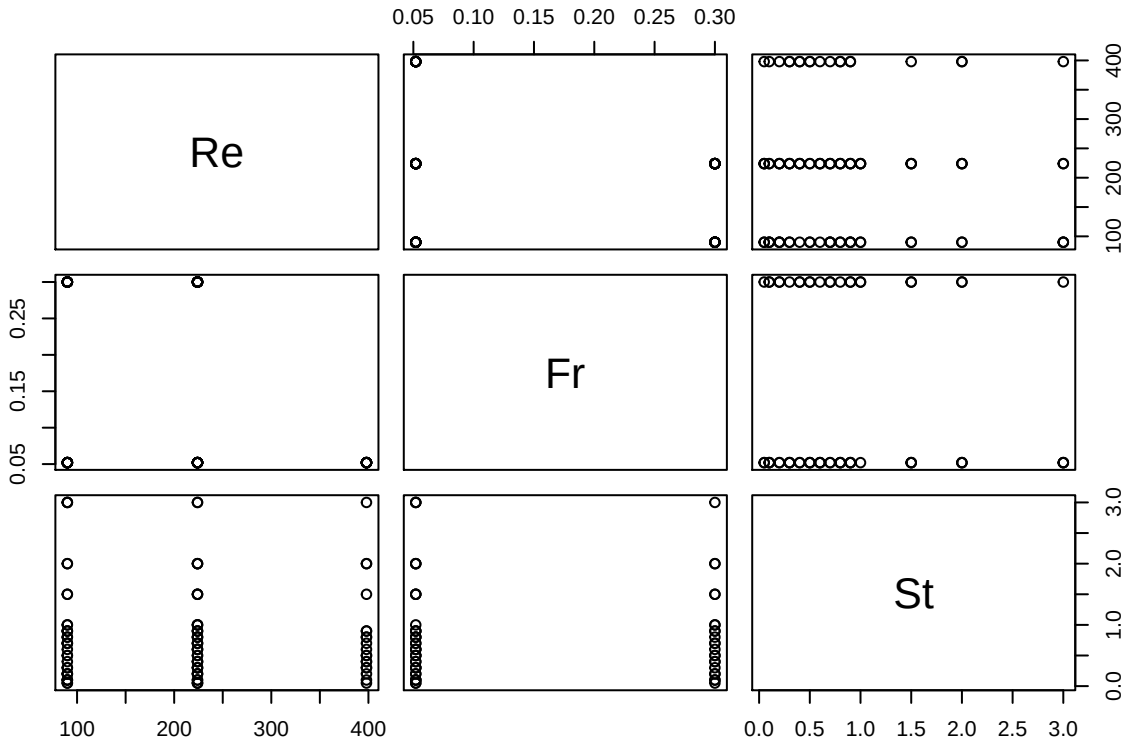
10/10/2020

```
st_m1 <- ggplot(train,aes(x=St, y=R_moment_1)) + geom_point()
re_m1 <- ggplot(train,aes(x=Re, y=R_moment_1)) + geom_point()
fr_m1 <- ggplot(train,aes(x=Fr, y=R_moment_1)) + geom_point()
```

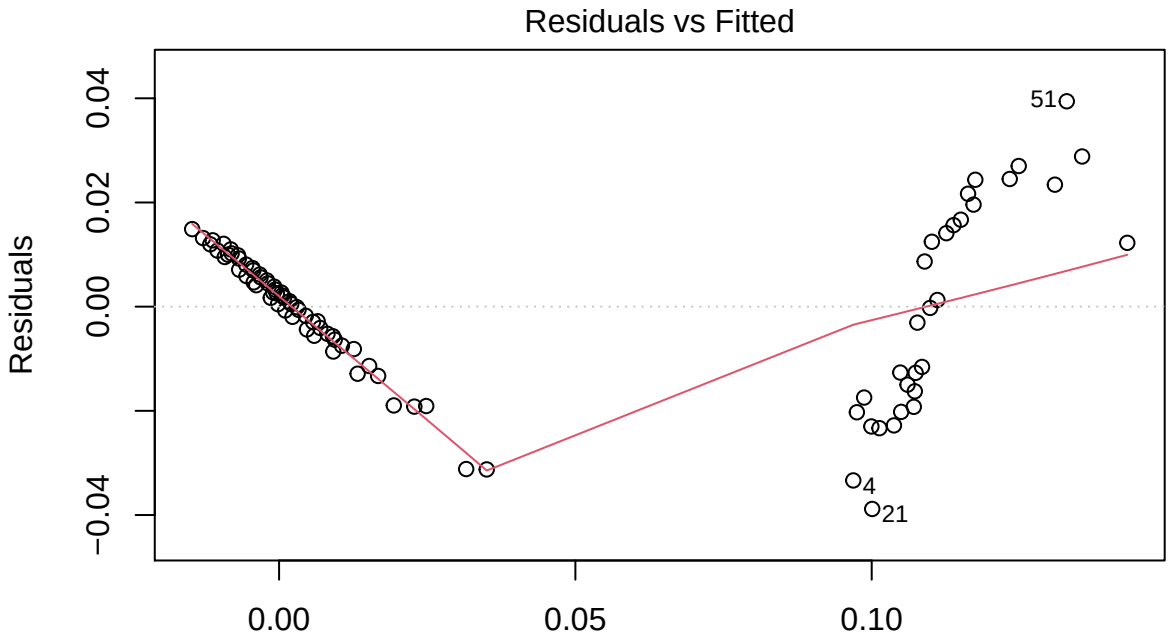
```
#pairs plot to show correlation between response variables
train_response <- data.frame(train[,c("R_moment_1","R_moment_2","R_moment_3","R_moment_4")])
pairs(train_response)
```

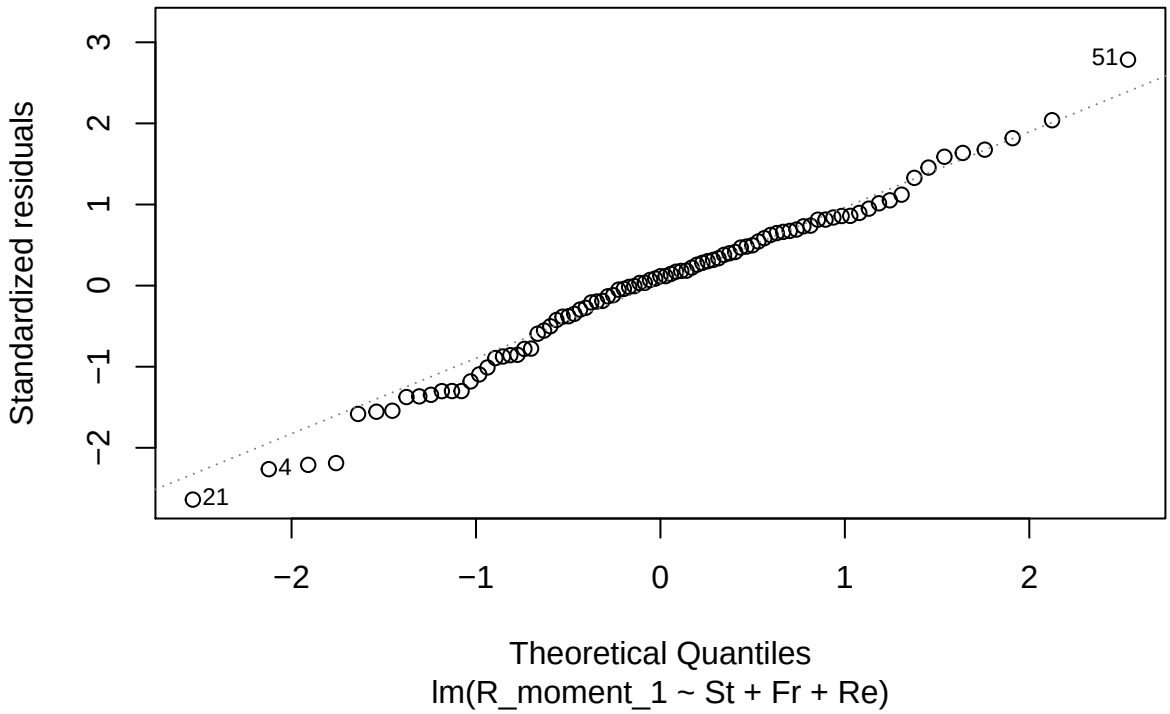


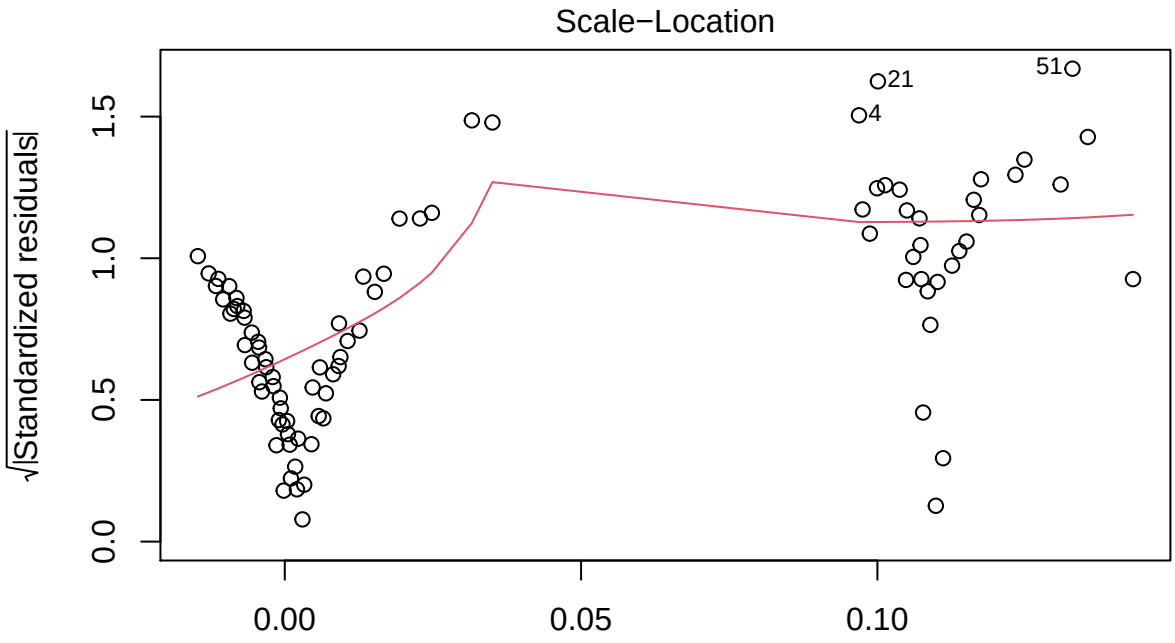

```
#correlation between predictor variables (in case we need to include interaction effects)
train_predictors <- data.frame(train[,c("Re","Fr","St")])
pairs(train_predictors)
```

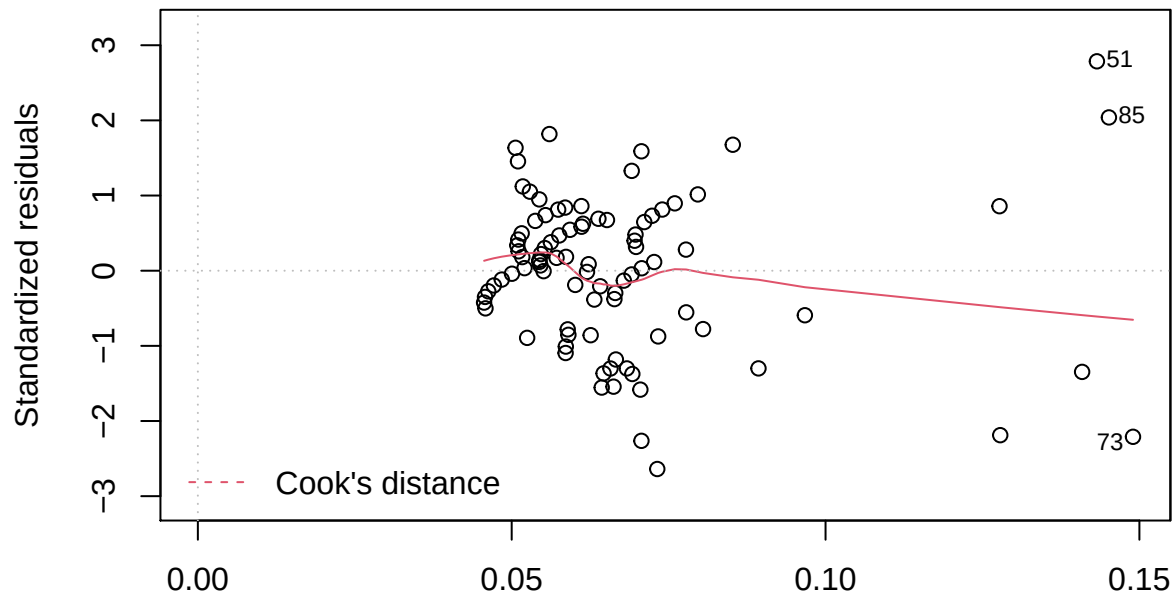



```r
#linear regression with factored Re and Fr
train1 <- train
train1$Re <- as.factor(train$Re)
train1$Fr <- as.factor(train$Fr)
lm1_m1 <- lm(R_moment_1 ~ St + Fr + Re, data=train1)
lm2_m1 <- lm(R_moment_2 ~ St + Fr + Re, data=train1)
lm3_m1 <- lm(R_moment_3 ~ St + Fr + Re, data=train1)
lm4_m1 <- lm(R_moment_4 ~ St + Fr + Re, data=train1)
plot(lm1_m1)
```


Residuals vs Fitted
Residuals
0.04
0.02
0.00
−0.04
51
4
21
0.00
0.05
0.10
Fitted values
lm(R_moment_1 ~ St + Fr + Re)


Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
51
4
21
−2
−1
0
1
2
Theoretical Quantiles
lm(R_moment_1 ~ St + Fr + Re)






```
x <- model.matrix(R_moment_1~St + Fr + Re,data=train_noinf)[,-1]
y <- train_noinf$R_moment_1

library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

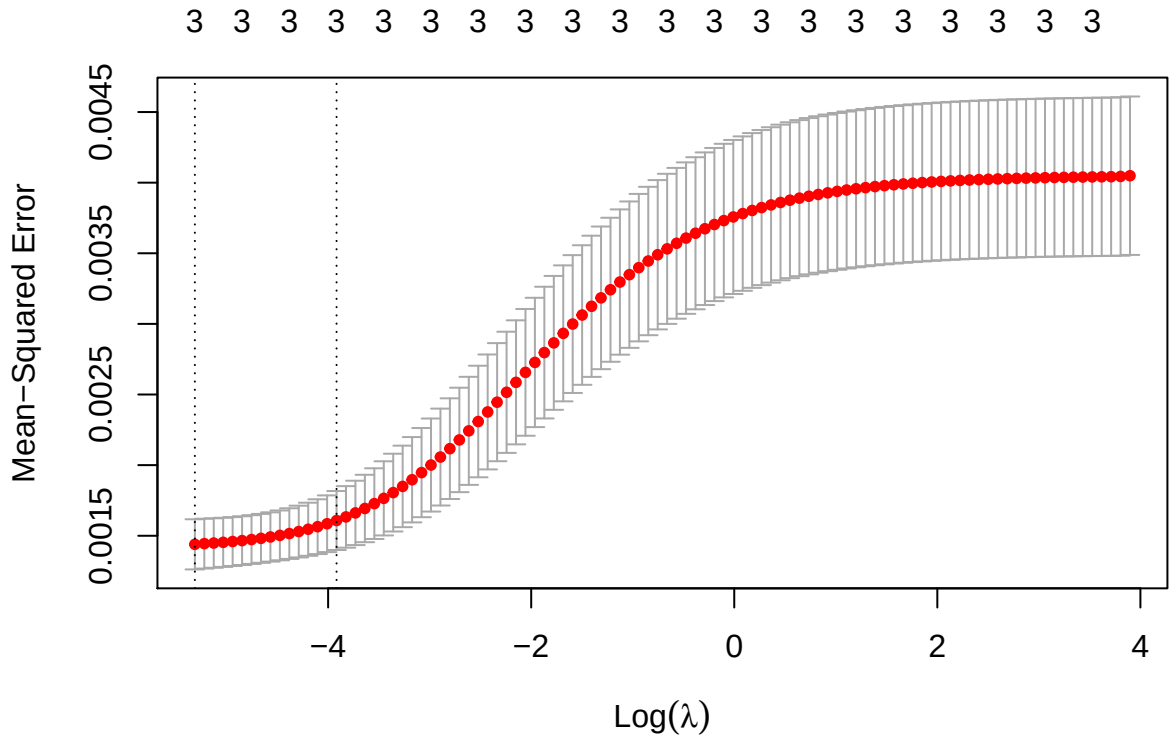
```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.0-2
```

```
grid <- 10^seq(10, -2, length = 100) # grid of values for lambda param
ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid)


train <- sample(1:nrow(x), nrow(x)/2)
test <- (-train)
y.test <- y[test]

ridge.mod <- glmnet(x[train,], y[train], alpha = 0, lambda = grid, thresh = 1e-12)
ridge.pred <- predict(ridge.mod, s = 4, newx = x[test,])
mean((ridge.pred - y.test)^2) ## calculate MSE
```

```
## [1] 0.002484908
```

```
set.seed(1)
cv.out <- cv.glmnet(x[train,], y[train], alpha = 0)
plot(cv.out)
```




```
bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.004929308
```

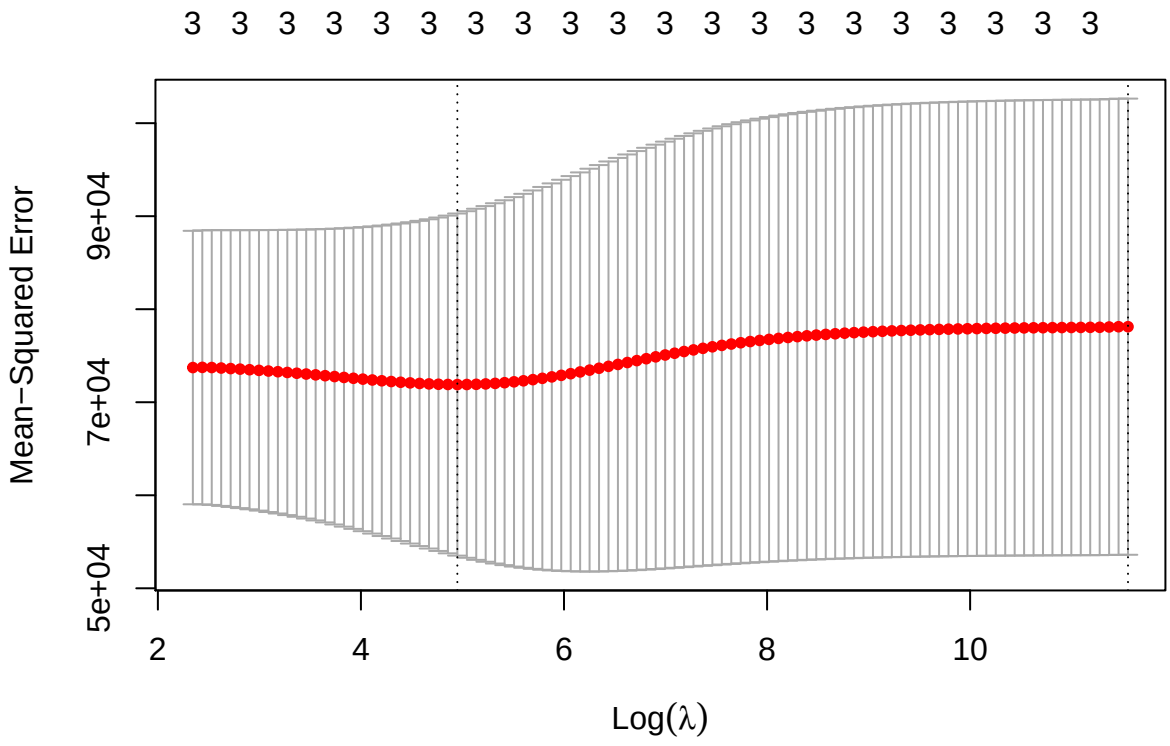
```
ridge.pred <- predict(ridge.mod, s = bestlam, newx = x[test,])
mean((ridge.pred - y.test)^2)
```

```
## [1] 0.001071318
```

```
x2 <- model.matrix(R_moment_2~ St + Fr + Re,data=train_noinf)[,-1]
y2 <- train_noinf$R_moment_2

library(glmnet)
set.seed(1)
grid <- 10^seq(10, -2, length = 100) # grid of values for lambda param
ridge.mod2 <- glmnet(x2, y2, alpha = 0, lambda = grid)


train2 <- sample(1:nrow(x2), nrow(x2)/2)
test2 <- (-train2)
y.test2 <- y2[test2]

ridge.mod2 <- glmnet(x2[train,], y2[train], alpha = 0, lambda = grid, thresh = 1e-12)
ridge.pred <- predict(ridge.mod, s = 4, newx = x2[test,])
mean((ridge.pred - y.test2)^2) ## calculate MSE
```

```
## [1] 66956.12
```

```
set.seed(1)
cv.out2 <- cv.glmnet(x2[train2,], y2[train2], alpha = 0)
plot(cv.out2)
```




```
bestlam2 <- cv.out2$lambda.min
bestlam2
```

```
## [1] 141.1089
```

```
ridge.pred2 <- predict(ridge.mod2, s = bestlam2, newx = x2[test2,])
mean((ridge.pred2 - y.test2)^2)
```

```
## [1] 44836.76
```

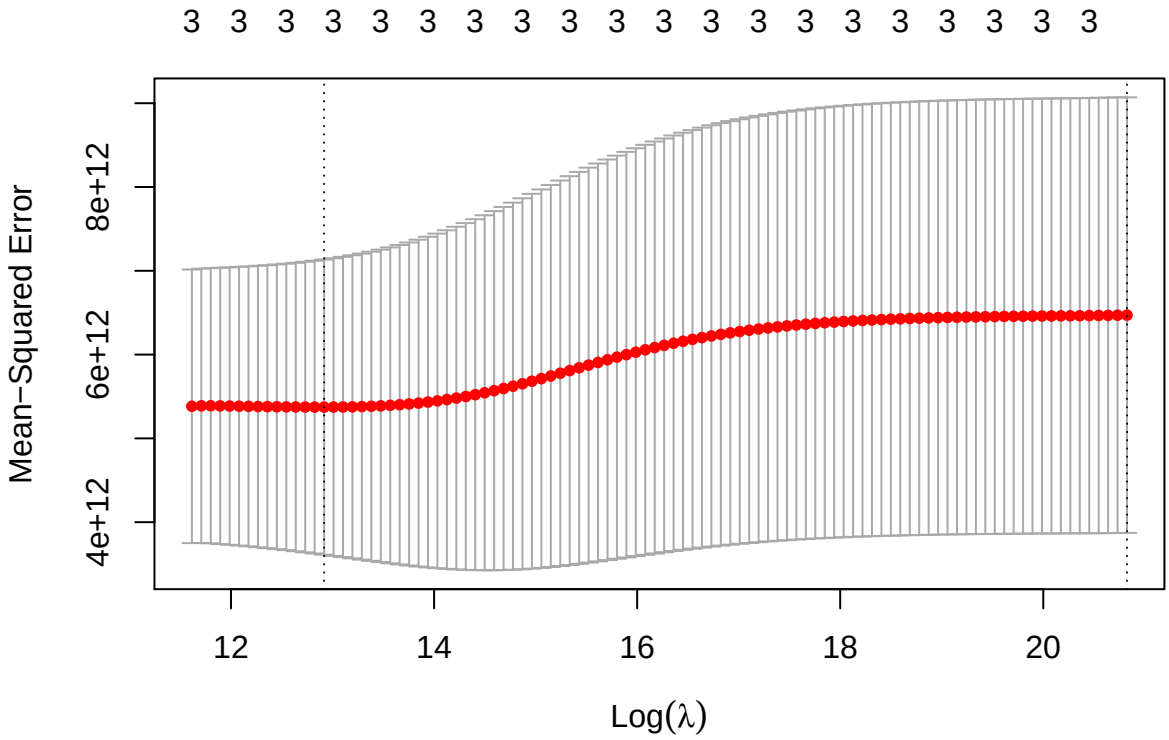
```
x3 <- model.matrix(R_moment_3~St + Fr + Re,data=train_noinf)[,-1]
y3 <- train_noinf$R_moment_3


ridge.mod3 <- glmnet(x3, y3, alpha = 0, lambda = grid)


train3 <- sample(1:nrow(x3), nrow(x3)/2)
test3 <- (-train3)
y.test3 <- y[test3]

ridge.mod3 <- glmnet(x3[train3,], y3[train3], alpha = 0, lambda = grid, thresh = 1e-12)
ridge.pred3 <- predict(ridge.mod3, s = 4, newx = x3[test3,])
mean((ridge.pred3 - y.test3)^2) ## calculate MSE
```

```
## [1] 2.924727e+12
```

```
x4 <- model.matrix(R_moment_4~St + Fr + Re,data=train_noinf)[,-1]
y4 <- train_noinf$R_moment_4

ridge.mod4 <- glmnet(x4, y4, alpha = 0, lambda = grid)


train4 <- sample(1:nrow(x4), nrow(x4)/2)
test4 <- (-train4)
y.test4 <- y[test4]

ridge.mod4 <- glmnet(x4[train4,], y4[train4], alpha = 0, lambda = grid, thresh = 1e-12)
ridge.pred4 <- predict(ridge.mod4, s = 4, newx = x4[test4,])
mean((ridge.pred4 - y.test4)^2) ## calculate MSE
```

```
## [1] 1.702767e+20
```

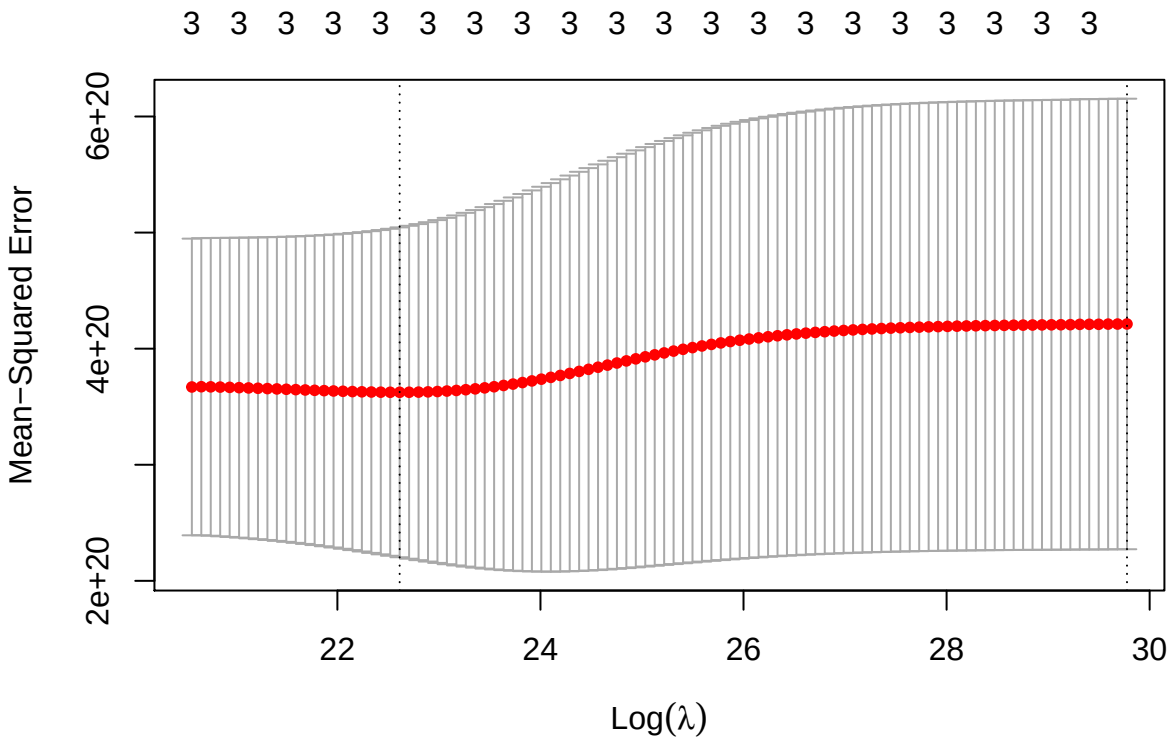
```
set.seed(1)
cv.out3 <- cv.glmnet(x3[train3,], y3[train3], alpha = 0)
plot(cv.out3)
```




```
bestlam3 <- cv.out3$lambda.min
bestlam3
```

```
## [1] 407052.8
```

```
cv.out4 <- cv.glmnet(x4[train4,], y4[train4], alpha = 0)
plot(cv.out4)
```



```
bestlam4 <- cv.out4$lambda.min
bestlam4
```

```
## [1] 6620092655
```

```
ridge.pred3 <- predict(ridge.mod3, s = bestlam3, newx = x3[test3,])
mean((ridge.pred3 - y.test3)^2)
```

```
## [1] 2.26338e+12
```

```
ridge.pred4 <- predict(ridge.mod4, s = bestlam4, newx = x4[test4,])
mean((ridge.pred4 - y.test4)^2)
```

```
## [1] 1.20962e+20
```

Introduction

Background Information

Turbulence is highly versatile motion that is often times difficult to predict and understanding fluid motion and its effect on natural problems poses a great challenge. Interpreting turbulence data is an incredibly important task in the engineering from understanding the cosmos to the cosmic cycle.

Parametric modeling is effective when we want to compactly represent features as model parameters. Unlike certain "black box" machine learning techniques, it offers a higher level of interpretability, which is especially useful for a practical setting.Rather than simply outputting a classification result, a parametric model allows us to investigate in more detail which aspects of turbulence differ between high and low particle cluster volumes.

In our project, we use linear and (. . .) and apply it to interpret the difference in model parameters in order to achieve the the following objectives:

1. Build a model that predict its particle cluster volume distribution in terms of the moments.
2. Investigate and interpret how model parameters affects the probability distribution for particle cluster volumes

Data

The data set procured for this case study consists of information about cluster volumes. In total, it contains 89 observations with 7 variables. Details of each variable is specified in Table 1.1. The original response variable, a probability distribution for particle cluster volumes, is difficult to interpret, and therefore is summarized into its first 4 raw moments.

Performing some exploratory data analysis, we can observe from Figure 1 that St, with the exception of the 0 values, follow a linear relationship with the first moment, while for Re, there is high variability for the first moment when $Re = 90$, and low variability otherwise. There is high variability across all Fr values with respect to the first moment. Some of these insights influenced our decisions for modelling approaches which will be further discussed in the Methods section.

Table 1: Table 1.1 Description Table of Data set variables

Metric	Value	Description
St	$0 < St < 3$	Particle property: effect on inertia (e.g. size, density)
Re	90, 224, 398	Reynolds Number: turbulent flow property
Fr	Infinity, 0, 3	Particle propety: gravitational acceleration
R moment 1	Continuous response variable	First raw moment of probability distribution
R moment 2	Continuous response variable	Second raw moment of volume probability distribution
R moment 3	Continuous response variable	Third raw moment of volume probability distribution
R moment 4	Continuous response variable	Fourth raw moment of volume probability distribution

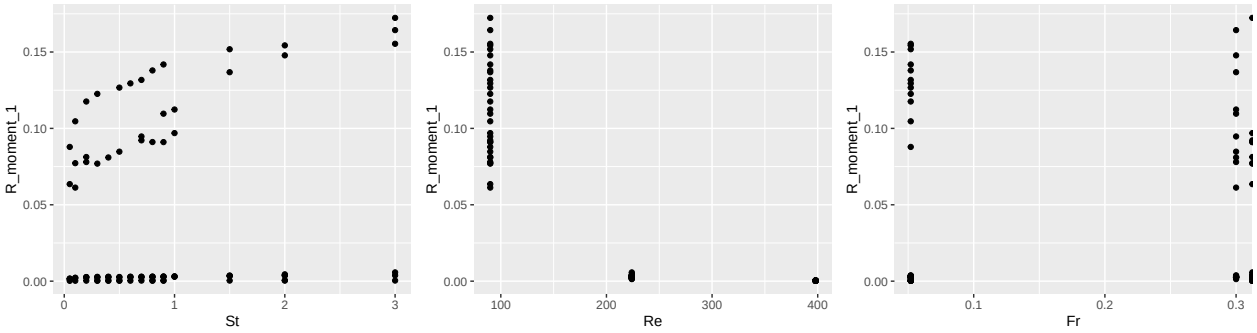


Figure 1: Exploratory data analysis plots on predictors vs first moment

Methodology

Results

Conclusion