Predicting Turbulence Simulations

Enzo Moraes Mescall, Martin Olarte, Charlotte Coudert

2022-10-31

```
data.train = read.csv("data-train.csv")
data.test = read.csv("data-test.csv")
```

Introduction

Methodology

```
clean = data.train %>%
  mutate(Fr = as.factor(Fr),
         Re = as.factor(Re))
```

After this, we created a function that would resemble our final product and take in a tuple of St, Re, and Fr and then return the four predicted moments. We reserved the global variable names `model_mX` for this final function.

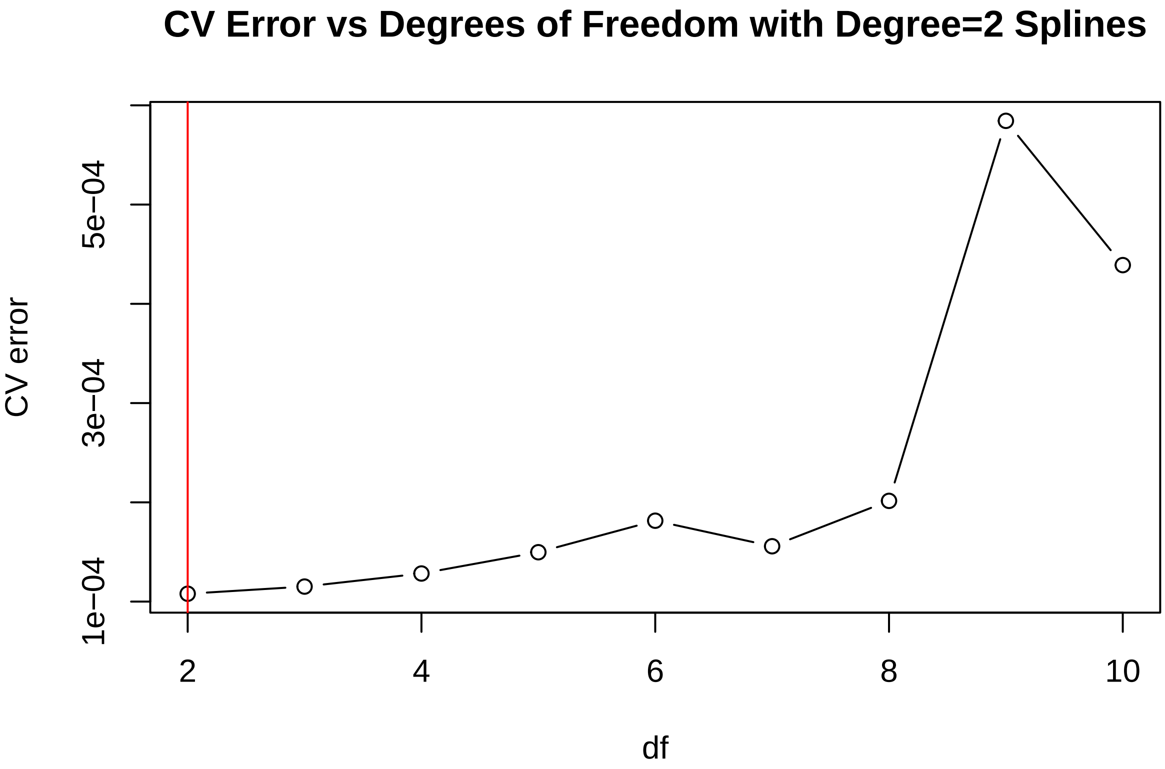
```
# temp models
predictive_model = function(test_St, test_Re, test_Fr) {
  newdata = data.frame(list(St = test_St,
                            Re = as.factor(test_Re),
                            Fr = as.factor(test_Fr)))

  m1 = predict(model_m1, newdata =  newdata)
  m2 = predict(model_m2, newdata =  newdata)
  m3 = predict(model_m3, newdata =  newdata)
  m4 = predict(model_m4, newdata =  newdata)

  return(c(m1, m2, m3, m4))
}
```

Since we have relatively few data points, n = 89, we decided to rely on LOOCV for our model selection process

```
errors = rep(NA, 10)
for (i in 2:10) {
  lm = glm(R_moment_1 ~ bs(St, df = i, degree = 2)*Re + Fr, data = clean)
  errors[i] = suppressWarnings(cv.glm(clean, lm)$delta[1])
}
plot(2:10, errors[-1], type="b", xlab="df", ylab="CV error",
     main = 'CV Error vs Degrees of Freedom with Degree=2 Splines')
abline(v = which.min(errors), col = "red")
```



Results

Conclusion