Eric.Rmd

Erie Seong Ho Han

10/11/2021

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(mgcv)
```

```
## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse

## This is mgcv 1.8-31. For overview type 'help("mgcv-package")'.
```

```r
test_data <- read.csv("data-test.csv")
train_data <- read.csv("data-train.csv")
```

```r
train_data <- train_data %>%
  mutate(Fr = as.factor(Fr)) %>%
  mutate(Re = as.factor(Re))

lm1 <- lm(R_moment_1 ~ St + Re + Fr, data = train_data)
gam1 <- gam(R_moment_1 ~ s(St) + Re + Fr, data = train_data)

gam2 <- gam(R_moment_1 ~ s(St, by = Fr) + Re + Fr, data = train_data)

gam3 <- gam(R_moment_1 ~ s(St, by = Re) + Re + Fr, data = train_data)
```

10-folds Cross validation skeleton code

```
# https://stats.stackexchange.com/questions/61090/how-to-split-a-data-set-to-do-10-fold-cross-validatio
set.seed(42)
# Randomly shuffle training data before splitting into 10 folds
shuffled_train <- train_data[sample(nrow(train_data)),]

# Create 10 folds
folds <- cut(seq(1,nrow(train_data)),breaks=10,labels=FALSE)

# error
rmse.cv.gam <- rep(0, 10)

# Cross validation: Use gam2 for example
for(i in 1:10){
    #Segement your data by fold using the which() function
    testIndexes <- which(folds==i,arr.ind=TRUE)
    testData <- shuffled_train[testIndexes, ]
    y.test <- testData$R_moment_1
    trainData <- shuffled_train[-testIndexes, ]

    #Use the test and train data
    gam_cv <- gam(R_moment_1 ~ s(St, by = Fr) + Re + Fr, data = trainData)
    pred_gam <- predict.gam(gam_cv, testData, type=’response’)

    rmse.cv.gam[i] = mean((pred_gam - y.test)^2)
}

print(mean(rmse.cv.gam)) # Estimated test error of gam from 10-folds CV (can do similarly for lm by att
```

```
## [1] 0.0002890787
```