1. Multiple y format (LR)

```
# read data --------------------------------------------------------------------

X_train <- read.csv("ModelData/X_train2.csv")[,-1]; Y_train <- read.csv("ModelData/Y_train2.csv")[,-1]
X_val <- read.csv("ModelData/X_val2.csv")[,-1]; Y_val <- read.csv("ModelData/Y_val2.csv")[,-1]
X_test <- read.csv("ModelData/X_test2.csv")[,-1]; Y_test <- read.csv("ModelData/Y_test2.csv")[,-1]

colnames(Y_train) <- c("Y1","Y2","Y3","Y4","Y5","Y6","Y7")
Data_train <- cbind(Y_train,X_train); #head(Data_train); dim(Data_train)

colnames(Y_val) <- c("Y1","Y2","Y3","Y4","Y5","Y6","Y7")
Data_val <- cbind(Y_val,X_val); #head(Data_val); dim(Data_val)

colnames(Y_test) <- c("Y1","Y2","Y3","Y4","Y5","Y6","Y7")
Data_test <- cbind(Y_test,X_test); #head(Data_test); dim(Data_test)
```

1.1. Linear regression without interactions

```
# LR ---------------------------------------------------------------------------

m1 <- lm(cbind(Data_train$Y1,Data_train$Y2,Data_train$Y3,Data_train$Y4,Data_train$Y5,Data_train$Y6,Data_train$Y7)
~ . , Data_train[,-c(1:7)])
#summary(m1)
#cat("MSE:", mean(m1$residuals^2))
#cat("RMSE:", sqrt(mean(m1$residuals^2)))

y_pred <- predict(m1, Data_val)
y_real <- Y_val
y_pred <- rbind(y_pred[,1],y_pred[,2],y_pred[,3],y_pred[,4],y_pred[,5],y_pred[,6],y_pred[,7])
y_real <- rbind(y_real[,1],y_real[,2],y_real[,3],y_real[,4],y_real[,5],y_real[,6],y_real[,7])
cat("MSE:", mean((y_real-y_pred)^2))
```

```
## MSE: 0.1117158
```

```
cat("RMSE:", sqrt(mean((y_real-y_pred)^2)))
```

```
## RMSE: 0.3342391
```

```
y_pred <- predict(m1, Data_test)
y_real <- Y_test
y_pred <- rbind(y_pred[,1],y_pred[,2],y_pred[,3],y_pred[,4],y_pred[,5],y_pred[,6],y_pred[,7])
y_real <- rbind(y_real[,1],y_real[,2],y_real[,3],y_real[,4],y_real[,5],y_real[,6],y_real[,7])
cat("MSE:", mean((y_real-y_pred)^2))
```

```
## MSE: 0.09493806
```

```
cat("RMSE:", sqrt(mean((y_real-y_pred)^2)))
```

```
## RMSE: 0.3081202
```

2. One y format, “all var & station” interactions (LR, LASSO, Ridge, Decision Tree)

```
# read data --------------------------------------------------------------------
Train <- read.csv("ModelData/Train2_long.csv"); #head(Train); dim(Train)
Val <- read.csv("ModelData/Val2_long.csv"); #head(Val); dim(Val)
Test <- read.csv("ModelData/Test2_long.csv"); #head(Test); dim(Test)

# categorical var
for(i in c(4:6)){
  Train[,i] <- as.character(Train[,i])
  Val[,i] <- as.character(Val[,i])
  Test[,i] <- as.character(Test[,i])
}
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
scaling_back <- group_by(Test, mrt_station)%>%
  summarise(median=median(original_mrt_flow),iqr=IQR(original_mrt_flow))
Test <- merge(Test,scaling_back)
Train <- Train[,-3]; Val <- Val[,-3]; Test <- Test[,-3]
#summary(Train); summary(Val); summary(Test)
```

2.1. Linear regression with interactions

```
lr_model <- lm(mrt_flow~ . * mrt_station, Train)
#lr_model <- lm(mrt_flow ~ . * mrt_station, rbind(Train,Val))
#summary(lr_model)
#cat("MSE:", mean(m1$residuals^2))
#cat("RMSE:", sqrt(mean(m1$residuals^2)))

y_pred <- predict(lr_model, Val)
```

```
## Warning in predict.lm(lr_model, Val): prediction from a rank-deficient fit may
## be misleading
```

```
y_real <- Val$mrt_flow
cat("MSE:", mean((y_real-y_pred)^2))
```

```
## MSE: 0.104414
```

```
cat("RMSE:", sqrt(mean((y_real-y_pred)^2)))
```

```
## RMSE: 0.3231316
```

```
y_pred <- predict(lr_model, Test[,-c(30:31)])
```

```
## Warning in predict.lm(lr_model, Test[, -c(30:31)]): prediction from a
## rank-deficient fit may be misleading
```

```
y_real <- Test$mrt_flow
cat("MSE:", mean((y_real-y_pred)^2))
```

```
## MSE: 0.06850735
```

```
cat("RMSE:", sqrt(mean((y_real-y_pred)^2)))
```

```
## RMSE: 0.2617391
```

```
cat("Original RMSE:", sqrt(mean(( (y_real*Test$iqr+Test$median) - (y_pred*Test$iqr+Test$median) )^2)))
```

```
## Original RMSE: 494.7286
```

2.2. LASSO with interactions

```
f <- as.formula(mrt_flow ~ .* mrt_station) # using .*. for all interactions
y <- Train$mrt_flow
x <- model.matrix(f, Train)[,-1] # using model.matrix to take advantage of f
#y <- rbind(Train,Val)$mrt_flow
#x <- model.matrix(f, rbind(Train,Val))[,-1] # using model.matrix to take advantage of f

library(glmnet)
```

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

```
## Loaded glmnet 4.1-7
```

```
lasso_kfold <- cv.glmnet(x, y, alpha=0, nfolds=10)
lasso_best_lambda <- lasso_kfold$lambda.min
lasso_model <- glmnet(x, y, alpha=0, lambda=lasso_best_lambda)

x <- model.matrix(mrt_flow ~.*mrt_station, rbind(Train,Val))[,-1][-(1:nrow(Train)),]
y_pred <- predict(lasso_model, x)
y_real <- Val$mrt_flow
cat("MSE:", mean((y_real-y_pred)^2))
```

```
## MSE: 0.1019242
```

```
cat("RMSE:", sqrt(mean((y_real-y_pred)^2)))
```

```
## RMSE: 0.3192557
```

```
x <- model.matrix(mrt_flow ~.*mrt_station, rbind(Train,Test[,-c(30:31)]))[,-1][-(1:nrow(Train)),]
#x <- model.matrix(mrt_flow ~.*mrt_station, rbind(Train,Val,Test))[,-1][-(1:nrow(rbind(Train,Val))),]
y_pred <- predict(lasso_model, x)
y_real <- Test$mrt_flow
cat("MSE:", mean((y_real-y_pred)^2))
```

```
## MSE: 0.07014739
```

```
cat("RMSE:", sqrt(mean((y_real-y_pred)^2)))
```

```
## RMSE: 0.2648535
```

```
cat("Original RMSE:", sqrt(mean(( (y_real*Test$iqr+Test$median) - (y_pred*Test$iqr+Test$median) )^2)))
```

```
## Original RMSE: 539.6949
```

2.3. Ridge with interactions

```
f <- as.formula(mrt_flow ~ .* mrt_station) # using .*. for all interactions
y <- Train$mrt_flow
x <- model.matrix(f, Train)[,-1] # using model.matrix to take advantage of f
#y <- rbind(Train,Val)$mrt_flow
#x <- model.matrix(f, rbind(Train,Val))[,-1] # using model.matrix to take advantage of f

library(glmnet)
ridge_kfold <- cv.glmnet(x, y, alpha=1, nfolds=10)
ridge_best_lambda <- ridge_kfold$lambda.min
ridge_model <- glmnet(x, y, alpha=1, lambda=ridge_best_lambda)

x <- model.matrix(mrt_flow ~.*mrt_station, rbind(Train,Val))[,-1][-(1:nrow(Train)),]
y_pred <- predict(ridge_model, x)
y_real <- Val$mrt_flow
cat("MSE:", mean((y_real-y_pred)^2))
```

```
## MSE: 0.100222
```

```
cat("RMSE:", sqrt(mean((y_real-y_pred)^2)))
```

```
## RMSE: 0.3165786
```

```
x <- model.matrix(mrt_flow ~.*mrt_station, rbind(Train,Test[,-c(30:31)]))[,-1][-(1:nrow(Train)),]
#x <- model.matrix(mrt_flow ~.*mrt_station, rbind(Train,Val,Test))[,-1][-(1:nrow(rbind(Train,Val))),]
y_pred <- predict(ridge_model, x)
y_real <- Test$mrt_flow
cat("MSE:", mean((y_real-y_pred)^2))
```

```
## MSE: 0.06638792
```

```
cat("RMSE:", sqrt(mean((y_real-y_pred)^2)))
```

```
## RMSE: 0.2576585
```

```
cat("Original RMSE:", sqrt(mean(( (y_real*Test$iqr+Test$median) - (y_pred*Test$iqr+Test$median) )^2)))
```

```
## Original RMSE: 481.8453
```

2.4. Elastic (combine LASSO and Ridge) with interactions

```
f <- as.formula(mrt_flow ~ . * mrt_station) # using .*. for all interactions
x <- model.matrix(f, Train)[,-1] # using model.matrix to take advantage of f
y <- Train$mrt_flow

library(glmnet)
set.seed(1)
elastic_model <- glmnet(x, y, alpha=0.05, lambda=0.01) # choose alpha and beta by the performance in val

x <- model.matrix(mrt_flow ~.*mrt_station, rbind(Train,Val))[,-1][-(1:nrow(Train)),]
y_pred <- predict(elastic_model, x)
y_real <- Val$mrt_flow
cat("MSE:", mean((y_real-y_pred)^2))
```

```
## MSE: 0.09878929
```

```
cat("RMSE:", sqrt(mean((y_real-y_pred)^2)))
```

```
## RMSE: 0.3143076
```

```
x <- model.matrix(mrt_flow ~.*mrt_station, rbind(Train,Test[,-c(30:31)]))[,-1][-(1:nrow(Train)),]
y_pred <- predict(elastic_model, x)
y_real <- Test$mrt_flow
cat("MSE:", mean((y_real-y_pred)^2))
```

```
## MSE: 0.06627837
```

```
cat("RMSE:", sqrt(mean((y_real-y_pred)^2)))
```

```
## RMSE: 0.2574459
```

```
cat("Original RMSE:", sqrt(mean(( (y_real*Test$iqr+Test$median) - (y_pred*Test$iqr+Test$median) )^2)))
```

```
## Original RMSE: 486.4082
```

2.5. Regression tree (decision tree)

```
library(rpart)
set.seed(1)
dt_model <- rpart(mrt_flow~., Train, cp=0.000004) # choose cp by the performance in val
#summary(dt_model)
#printcp(dt_model); plotcp(dt_model)
dt_model$variable.importance
```

```
##         bike_flow              hour previous_mrt_flow                no
##       21736.49162       21515.46961       12988.46256        1768.69502
##       mrt_station               nox       day_in_a_week               no2
##        1704.84437        1648.53914        1245.69957        1163.52528
##             month            o3_8hr   air_temperature      air_pressure
##         716.73295         522.52496         341.15955         335.46342
## relative_humidity                o3            co_8hr                co
##         283.58198         256.31721         174.65703         124.11705
## sunshine_duration         windspeed          winddirec               aqi
##          94.74418          93.77009          93.05271          85.13969
##         pm2.5_avg           so2_avg     precipitation          pm10_avg
##          48.81502          44.62035          43.77014          42.35497
##              pm10               so2             pm2.5            status
##          41.24490          24.42436          23.61094          19.64605
```

```
setdiff((dt_model$frame$var), "<leaf>") # variables used
```

```
##  [1] "bike_flow"         "hour"              "mrt_station"
##  [4] "previous_mrt_flow" "day_in_a_week"     "air_pressure"
##  [7] "aqi"               "relative_humidity" "month"
## [10] "no2"               "pm2.5"             "pm10"
## [13] "no"                "pm10_avg"          "nox"
## [16] "o3_8hr"            "winddirec"         "sunshine_duration"
## [19] "precipitation"     "air_temperature"   "co"
## [22] "o3"                "so2"               "windspeed"
## [25] "co_8hr"            "so2_avg"           "pm2.5_avg"
```

```
y_pred <- predict(dt_model, Val)
y_real <- Val$mrt_flow
cat("MSE:", mean((y_real-y_pred)^2))
```

```
## MSE: 0.05324942
```

```
cat("RMSE:", sqrt(mean((y_real-y_pred)^2)))
```

```
## RMSE: 0.2307584
```

```
y_pred <- predict(dt_model, Test[,-c(30:31)])
y_real <- Test$mrt_flow
cat("MSE:", mean((y_real-y_pred)^2))
```

```
## MSE: 0.02402419
```

```
cat("RMSE:", sqrt(mean((y_real-y_pred)^2)))
```

```
## RMSE: 0.1549974
```

```
cat("Original RMSE:", sqrt(mean(( (y_real*Test$iqr+Test$median) - (y_pred*Test$iqr+Test$median) )^2)))
```

```
## Original RMSE: 301.4344
```