random_forest_stream_T

Tao

2021-12-16

load the packages

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
## Loading required package: sp
##
## Attaching package: 'raster'
## The following object is masked from 'package:dplyr':
##
##     select
## Checking rgeos availability: TRUE
##
## Attaching package: 'EnvStats'
## The following objects are masked from 'package:moments':
##
##     kurtosis, skewness
## The following object is masked from 'package:baseflow':
##
##     print
## The following objects are masked from 'package:raster':
##
##     cv, predict, print
## The following objects are masked from 'package:stats':
##
##     predict, predict.lm
```

```
## The following object is masked from 'package:base':
##
##     print.default
```

```
## Registered S3 method overwritten by 'hoardr':
##   method           from
##   print.cache_info httr
```

```
##
## Attaching package: 'pracma'
```

```
## The following object is masked from 'package:purrr':
##
##     cross
```

```
## Loading required package: proto
```

```
## Loading required package: randomForest
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

#load the data

```
#the mean Aug stream temp at the USGS sites
#load("meanAugT_df.Rdata")
#the the PRISM air temperature data
#load("temp_all.Rdata")
# merged air and stream T
load("meanAugT_all.Rdata")
```

calculate monthly temp

```
colnames(meanAugT_all)<-c(colnames(meanAugT_all)[1:4],"Daily_Stream_T","X_00010_00003_cd", "Daily_Q", "

meanAugT_all_mo<-meanAugT_all %>%
  group_by(site_no  ,yr) %>%
  summarise(monthly_stream_T = mean(Daily_Stream_T), monthly_stream_Q = mean(Daily_Q),ele= mean(ele), m
```

```
## `summarise()` has grouped output by 'site_no'. You can override using the `.groups` argument.
```

```
head(meanAugT_all_mo)
```

```
## # A tibble: 6 x 6
## # Groups:   site_no [1]
##   site_no  yr    monthly_stream_T monthly_stream_Q   ele monthly_air_T
##   <chr>    <chr>            <dbl>            <dbl> <dbl>         <dbl>
## 1 10396000 2013              18.5             30.3  4254          20.6
```

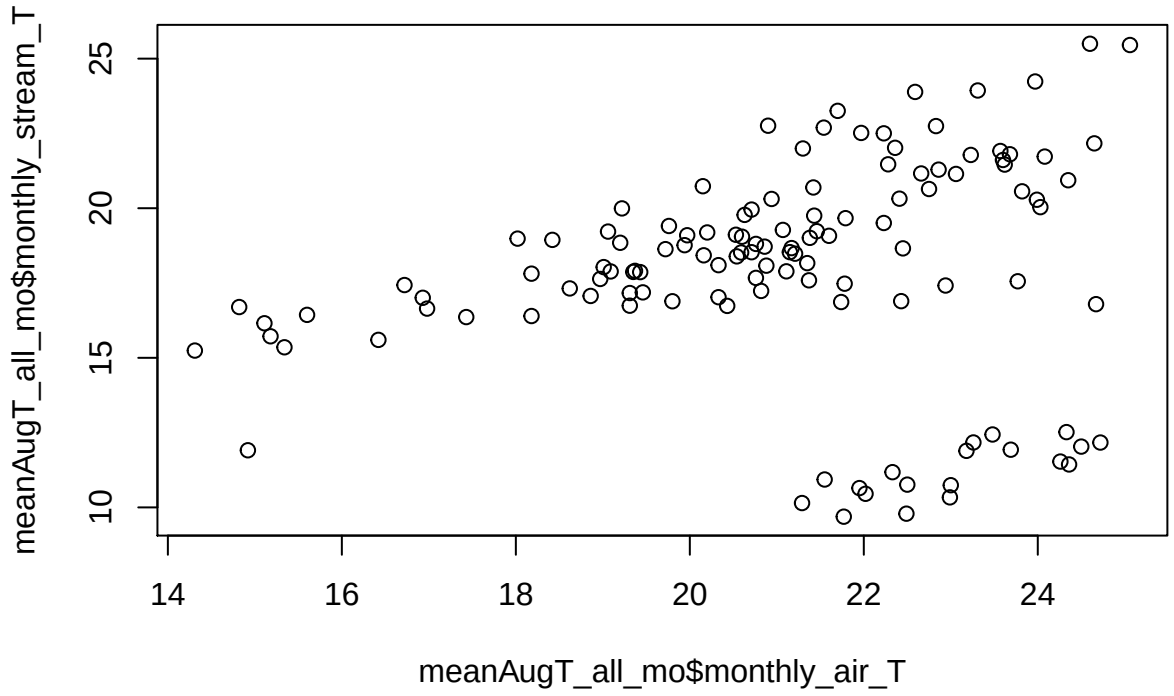
```
## 2 10396000 2014             18.6             28.9  4254          19.7
## 3 10396000 2015             18.8             26.8  4254          20.8
## 4 10396000 2016             18.4             35.0  4254          20.5
## 5 10396000 2017             18.7             40.4  4254          22.5
## 6 10396000 2018             18.5             25.3  4254          20.7
```

split the data into ref and non-ref streams

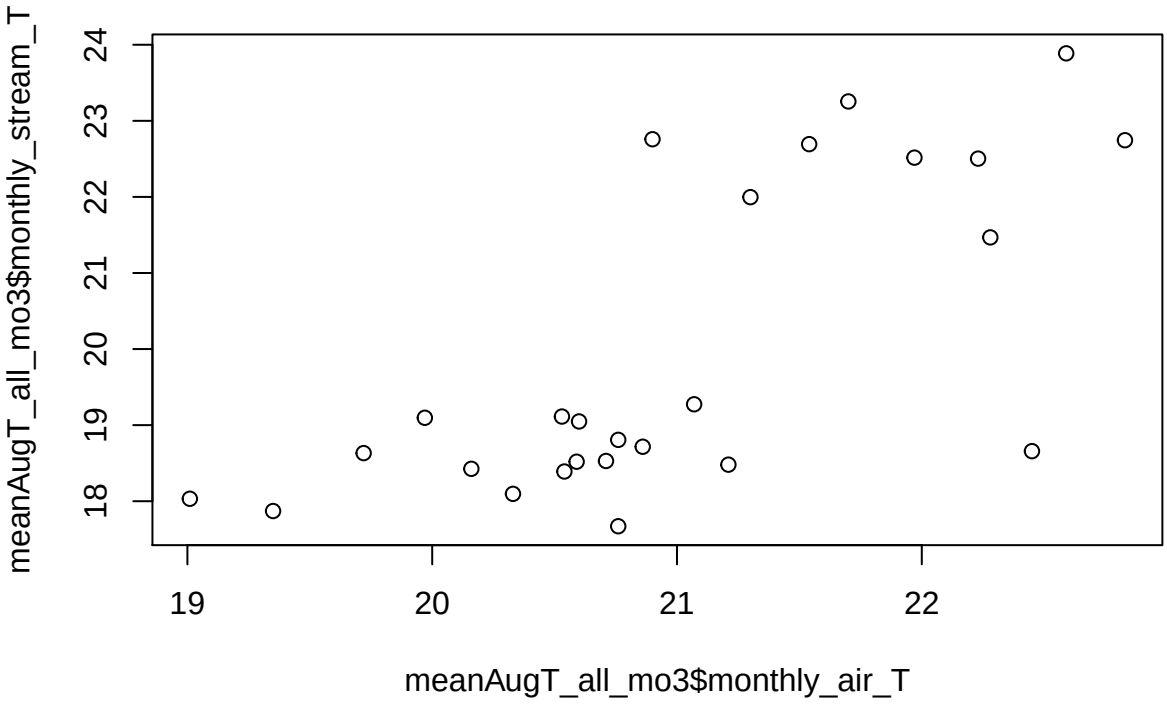
```
#g<-read.dbf("C:/Users/taohuang/Documents/Tao/Data/gagesII_9322_point_shapefile/gagesII_9322_sept30_201
g<-read.dbf("gagesII_9322_sept30_2011.dbf")

g$STAID<-as.character(g$STAID)
meanAugT_all_mo2<-merge(g,meanAugT_all_mo,by.x="STAID",by.y="site_no")
meanAugT_all_mo3<-meanAugT_all_mo2[meanAugT_all_mo2$CLASS=="Ref",]
meanAugT_all_mo4<-meanAugT_all_mo2[meanAugT_all_mo2$CLASS=="Non-ref",]
```

#plot monthly air T vs. stream T

```
plot(meanAugT_all_mo$monthly_air_T, meanAugT_all_mo$monthly_stream_T)
```




```
plot(meanAugT_all_mo3$monthly_air_T,meanAugT_all_mo3$monthly_stream_T)
```



Generate training and test data

```
# I splitted dataset into training and test data. The test data will be 30% of the entire dataset.
set.seed(101)
train = sample(1:nrow(meanAugT_all_mo),  nrow(meanAugT_all_mo)*0.7 )
dim(meanAugT_all_mo)
```

```
## [1] 124   6
```

```
length(train)
```

```
## [1] 86
```

```
set.seed(101)
train2 = sample(1:nrow(meanAugT_all_mo2),  nrow(meanAugT_all_mo2)*0.7 )
dim(meanAugT_all_mo2)
```

```
## [1] 124  19
```

```
length(train2)
```

```
## [1] 86
```

```
set.seed(101)
train3 = sample(1:nrow(meanAugT_all_mo3),  nrow(meanAugT_all_mo3)*0.7 )
dim(meanAugT_all_mo3)
```

```
## [1] 26 19
```

```
length(train3)
```

```
## [1] 18
```

```
set.seed(101)
train4 = sample(1:nrow(meanAugT_all_mo4),  nrow(meanAugT_all_mo4)*0.7 )
dim(meanAugT_all_mo4)
```

```
## [1] 98 19
```

```
length(train4)
```

```
## [1] 68
```

#run the random forest models

```
rf.stream_T = randomForest(monthly_stream_T ~ ele +monthly_stream_Q   +monthly_air_T , data = meanAugT_a
rf.stream_T
```

```
##
## Call:
##  randomForest(formula = monthly_stream_T ~ ele + monthly_stream_Q +      monthly_air_T, data = meanAu
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 1.608135
##                     % Var explained: 88.98
```

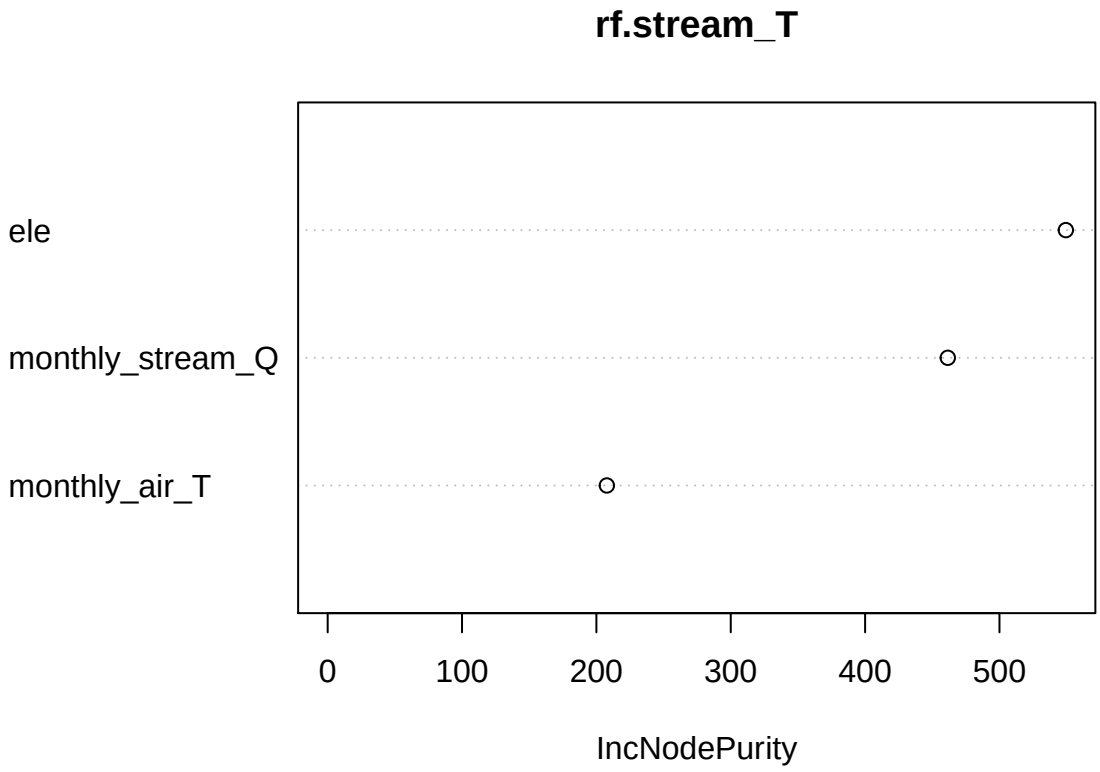
```
rf.stream_T$importance
```

```
##                   IncNodePurity
## ele                    549.3844
## monthly_stream_Q       461.5235
## monthly_air_T          207.7801
```

```
randomForest::varImpPlot(rf.stream_T, importance=TRUE)
```

```
## Warning in mtext(labs, side = 2, line = loffset, at = y, adj = 0, col = color, :
## "importance" is not a graphical parameter
```

```
## Warning in title(main = main, xlab = xlab, ylab = ylab, ...): "importance" is
## not a graphical parameter
```




```
rf.stream_T2 = randomForest(monthly_stream_T ~ ele +monthly_stream_Q   +monthly_air_T +CLASS , data = m
rf.stream_T2
```

```
##
## Call:
##  randomForest(formula = monthly_stream_T ~ ele + monthly_stream_Q +      monthly_air_T + CLASS, data
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 2.298078
##                     % Var explained: 84.25
```

```
rf.stream_T2$importance
```

```
##                   IncNodePurity
## ele                   450.11257
## monthly_stream_Q      359.19833
## monthly_air_T         149.36990
## CLASS                  75.16233
```

```
rf.stream_T3 = randomForest(monthly_stream_T ~ ele +monthly_stream_Q   +monthly_air_T , data = meanAugT
rf.stream_T3
```

```
##
## Call:
##  randomForest(formula = monthly_stream_T ~ ele + monthly_stream_Q +      monthly_air_T, data = meanAu
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
```

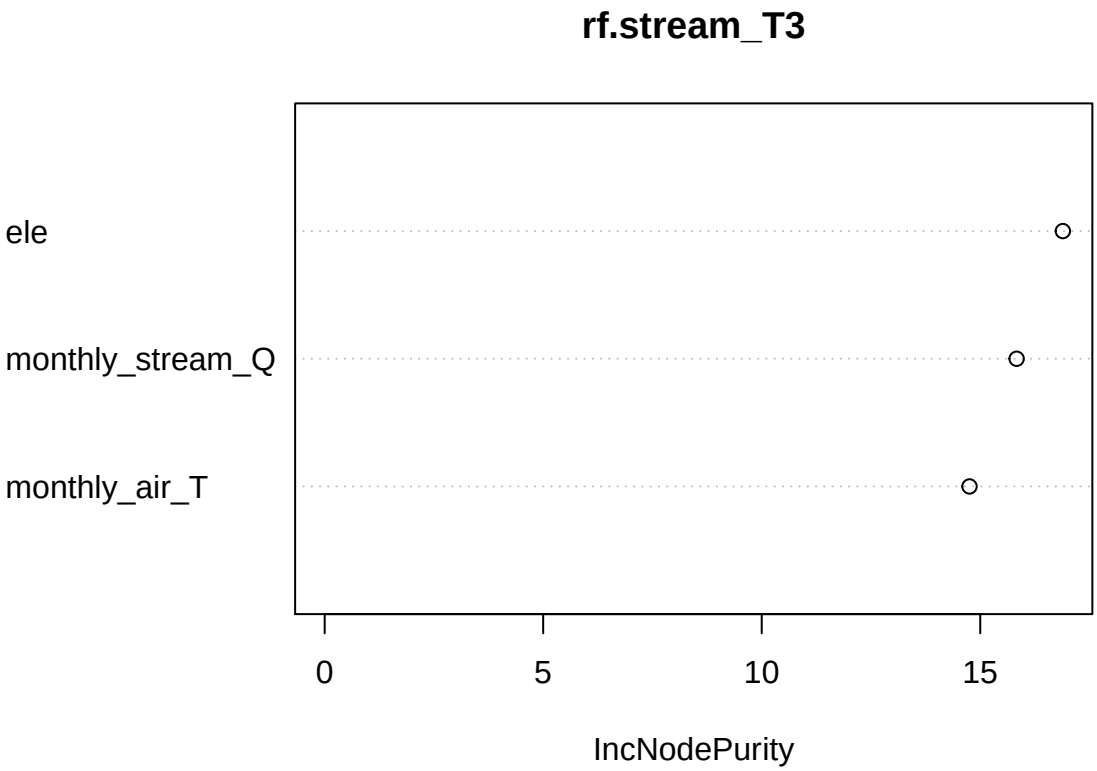
```
##           Mean of squared residuals: 0.4276459
##                     % Var explained: 85.81
```

```
rf.stream_T3$importance
```

```
##                   IncNodePurity
## ele                     16.88937
## monthly_stream_Q        15.83215
## monthly_air_T           14.75399
```

```
randomForest::varImpPlot(rf.stream_T3, importance=TRUE)
```

```
## Warning in mtext(labs, side = 2, line = loffset, at = y, adj = 0, col = color, :
## "importance" is not a graphical parameter

## Warning in mtext(labs, side = 2, line = loffset, at = y, adj = 0, col = color, :
## "importance" is not a graphical parameter
```




```
rf.stream_T4 = randomForest(monthly_stream_T ~ ele +monthly_stream_Q   +monthly_air_T , data = meanAugT
rf.stream_T4
```

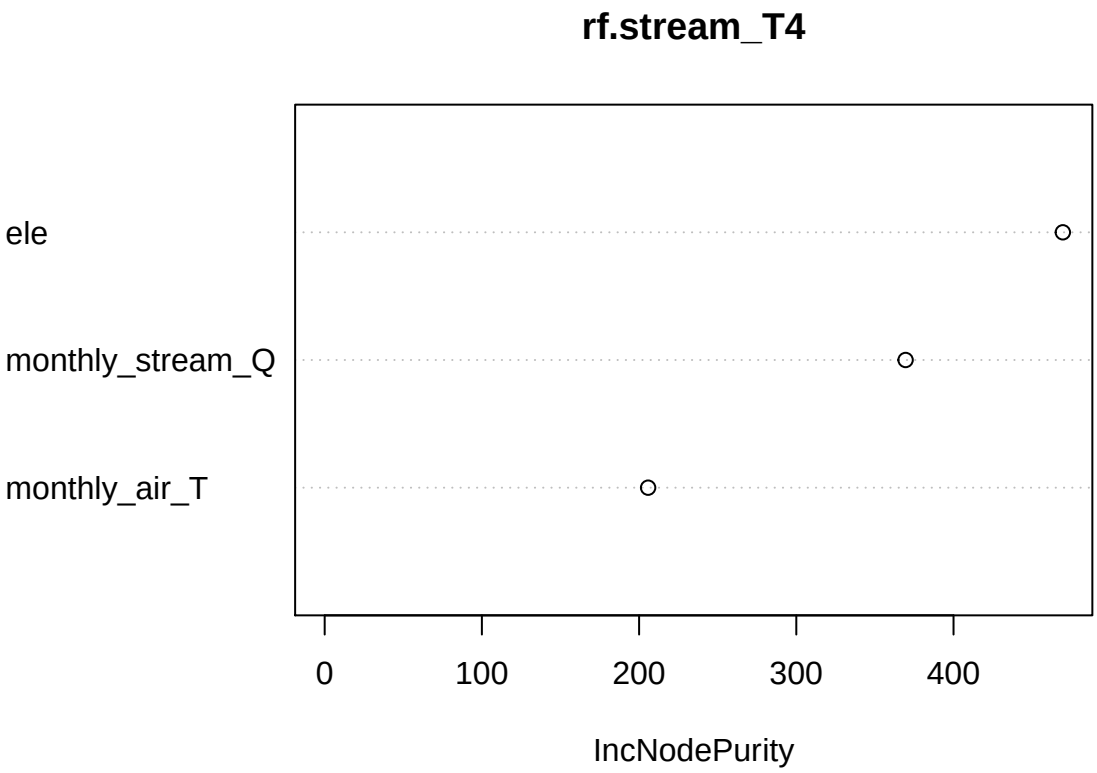
```
##
## Call:
##  randomForest(formula = monthly_stream_T ~ ele + monthly_stream_Q +      monthly_air_T, data = meanAu
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 3.49533
##                     % Var explained: 78.42
```

```
rf.stream_T4$importance
```

```
##                   IncNodePurity
## ele                     469.5243
## monthly_stream_Q        369.5098
## monthly_air_T           205.7340
```

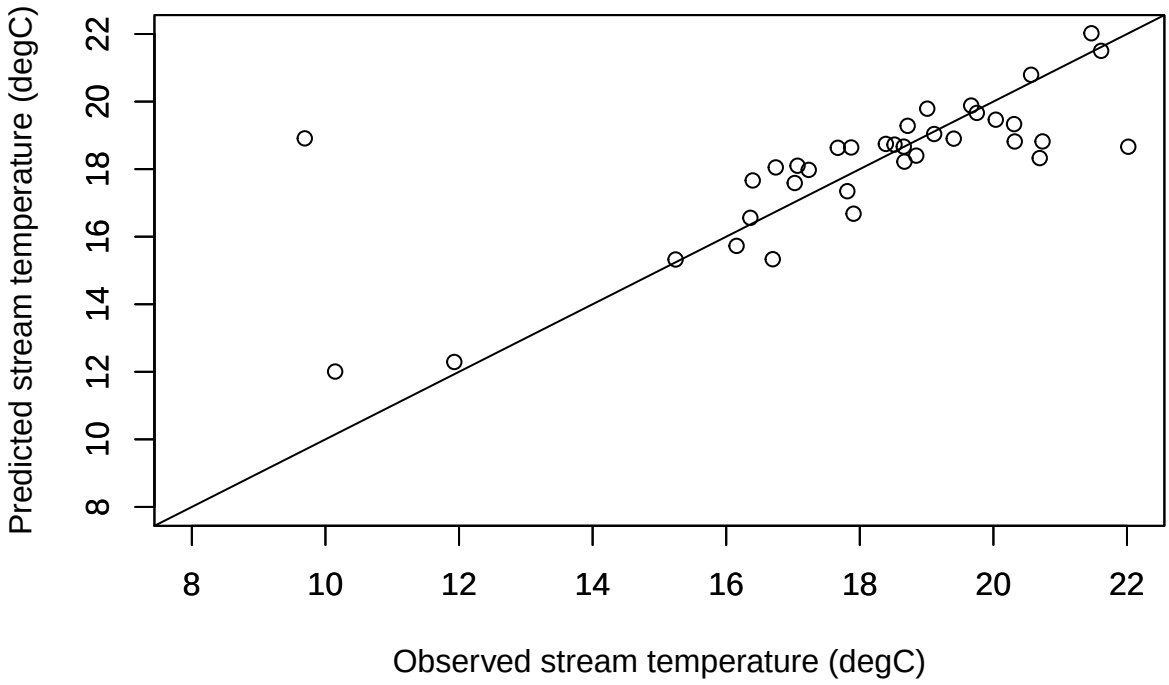
```
randomForest::varImpPlot(rf.stream_T4, importance=TRUE)
```

```
## Warning in mtext(labs, side = 2, line = loffset, at = y, adj = 0, col = color, :
## "importance" is not a graphical parameter

## Warning in mtext(labs, side = 2, line = loffset, at = y, adj = 0, col = color, :
## "importance" is not a graphical parameter
```



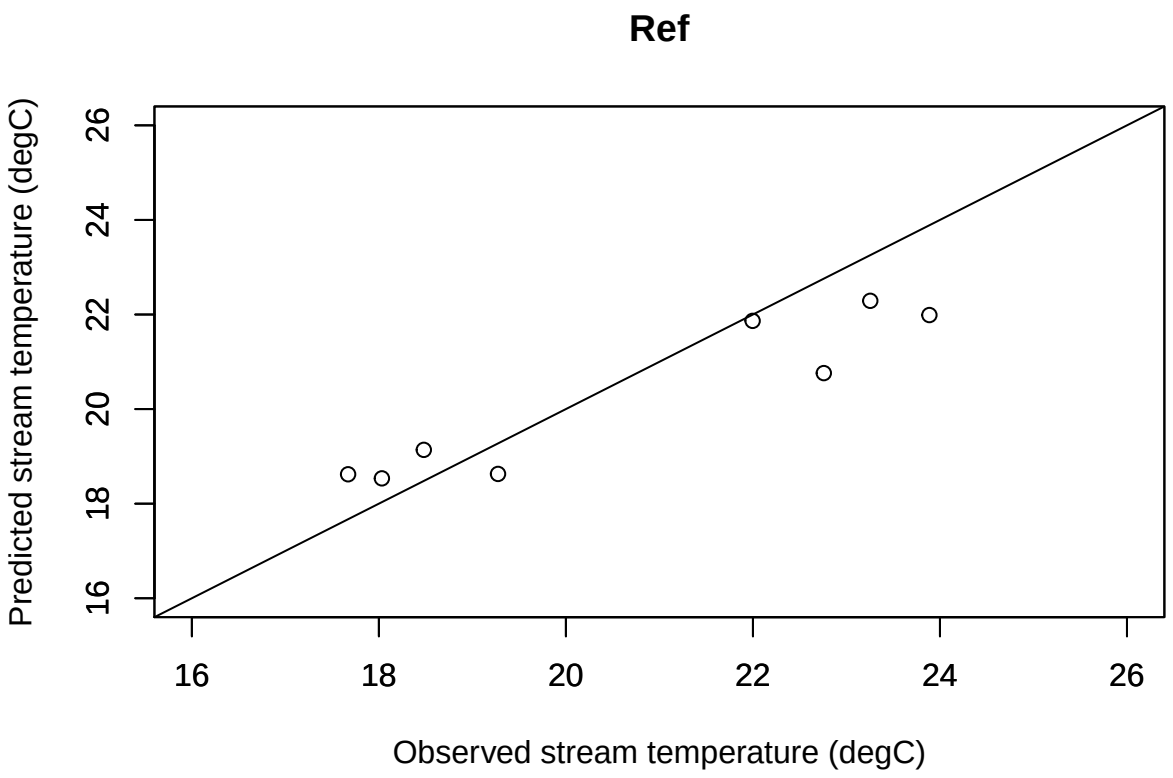
Make prediction of the all model

```
predictions <- predict(rf.stream_T, meanAugT_all_mo[-train,] )

x=seq(1,30)

plot(meanAugT_all_mo[-train,]$monthly_stream_T,predictions,xlim=c(8,22) ,ylim=c(8,22), xlab="Observed s
par(new=T)
plot(x,x,type="l",xlim=c(8,22) ,ylim=c(8,22),xlab="",ylab="")
```



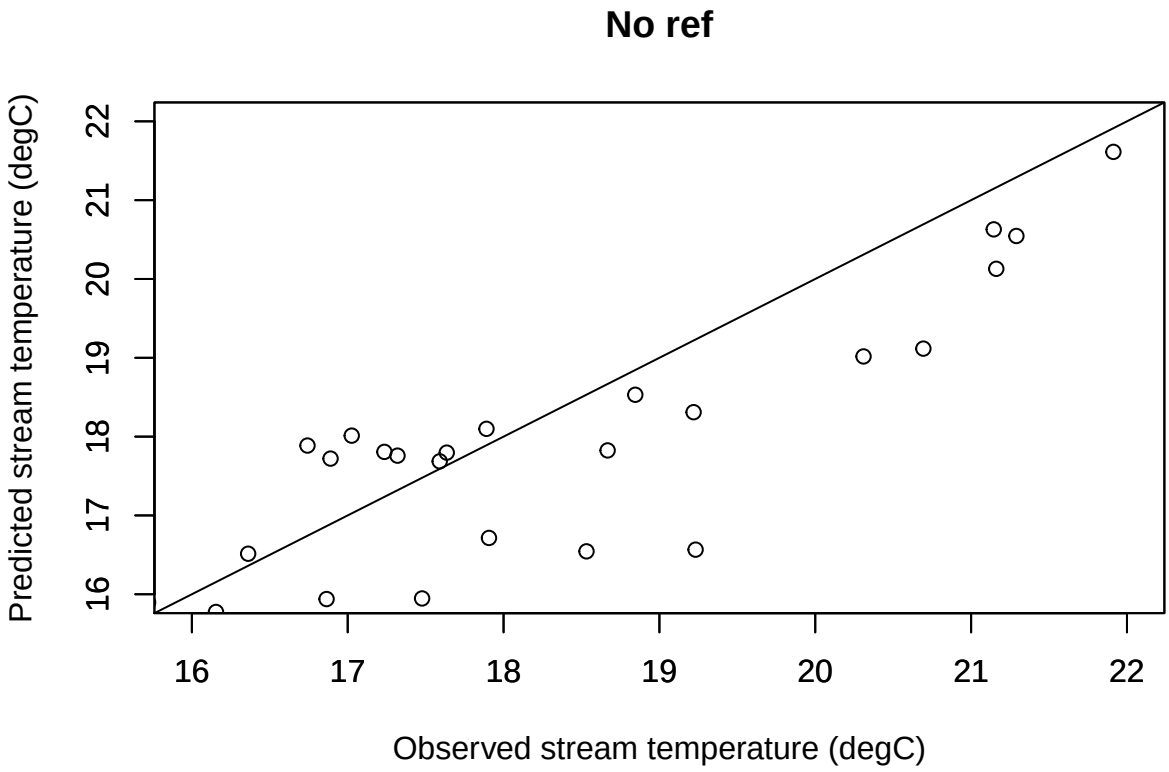
Make prediction of the ref model

```
predictions3 <- predict(rf.stream_T3, meanAugT_all_mo3[-train3,] )

plot(meanAugT_all_mo3[-train3,]$monthly_stream_T,predictions3 ,xlim=c(16,26) ,ylim=c(16,26) , xlab="Obs

par(new=T)
plot(x,x,type="l" ,xlim=c(16,26) ,ylim=c(16,26) ,xlab="",ylab="")
```



Make prediction of the non-ref model

```
predictions4 <- predict(rf.stream_T4, meanAugT_all_mo4[-train4,] )

plot(meanAugT_all_mo4[-train4,]$monthly_stream_T,predictions4 ,xlim=c(16,22) ,ylim=c(16,22) , xlab="Obs
par(new=T)
plot(x,x,type="l" ,xlim=c(16,22) ,ylim=c(16,22) ,xlab="",ylab="")
```


No ref
Predicted stream temperature (degC)
16
17
18
19
20
21
22
Observed stream temperature (degC)