models

Tao

2023-01-20

```
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
```

```
#install.packages('corrplot')
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(RColorBrewer)
# install.packages("gbm")
library("gbm")
```

```
## Loaded gbm 2.1.8
```

```
# install.packages("caret")
library("caret")
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
#install.packages("pdp")
library("pdp")          # model visualization
library("ggplot2")      # model visualization
#install.packages("lime")
library("lime")         # model visualization
```

```
##
## Attaching package: 'lime'
```

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

```
library("pROC")
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
```

```
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
#install.packages("e1071", repos="http://R-Forge.R-project.org")
library("e1071")
library( "MASS" )    #    used to generate correlated variables
```

```
##
## Attaching package: 'MASS'

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

```
library("sp")
library("Hmisc")    #   used for graphing se bars
```

```
## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following object is masked from 'package:e1071':
##
##     impute

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
#install.packages("randomForest")
require("randomForest")
```

```
## Loading required package: randomForest

## randomForest 4.7-1.1

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

##
## Attaching package: 'randomForest'

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

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

```
#install.packages("e1071")
library(e1071)
library(caret)
library("ModelMetrics")
```

```
##
## Attaching package: 'ModelMetrics'
```

```
## The following object is masked from 'package:pROC':
##
##     auc
```

```
## The following objects are masked from 'package:caret':
##
##     confusionMatrix, precision, recall, sensitivity, specificity
```

```
## The following object is masked from 'package:base':
##
##     kappa
```

```
library("foreign")
#install.packages("rfUtilities")
library("rfUtilities")
```

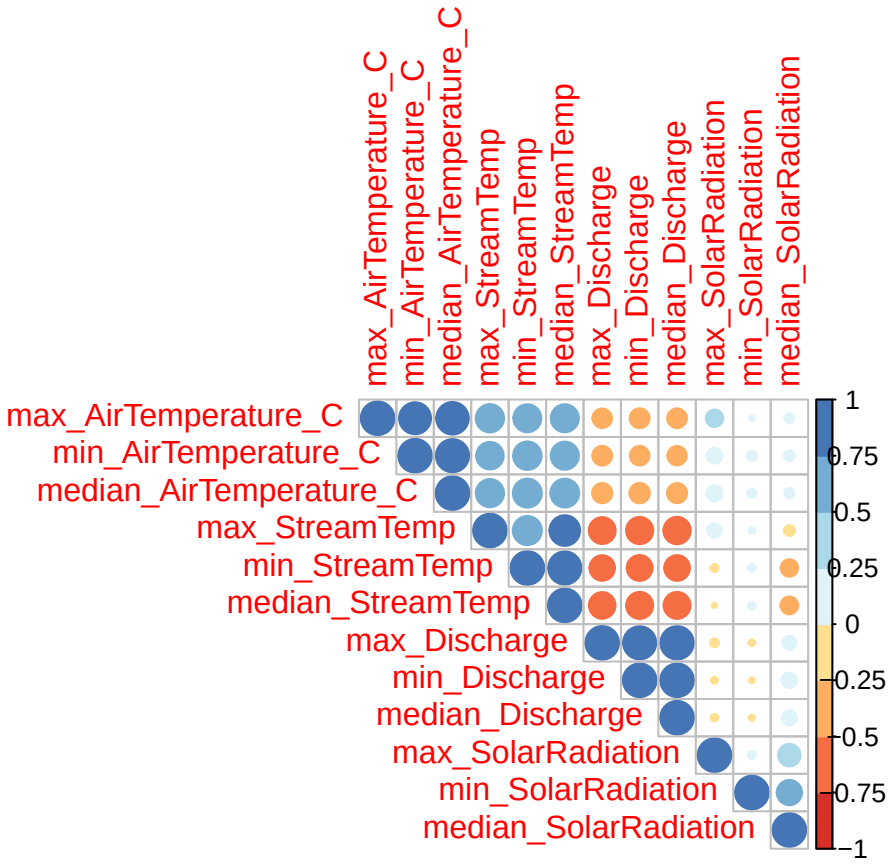
```
##
## Attaching package: 'rfUtilities'
```

```
## The following object is masked from 'package:ModelMetrics':
##
##     logLoss
```

Load data

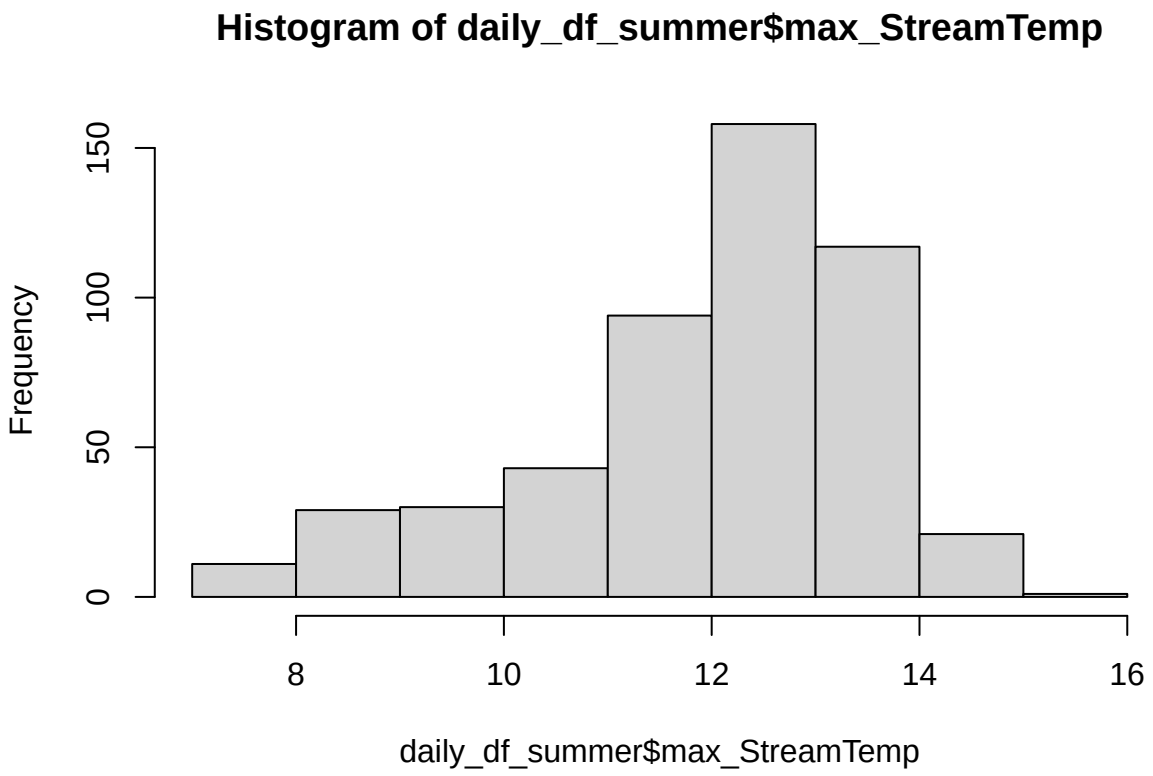
```
load("daily_df_summer.Rdata")
```

```
M <-cor(daily_df_summer[,c(2:13)])
corrplot(M, type="upper", order="hclust",
         col=brewer.pal(n=8, name="RdYlBu"))
```

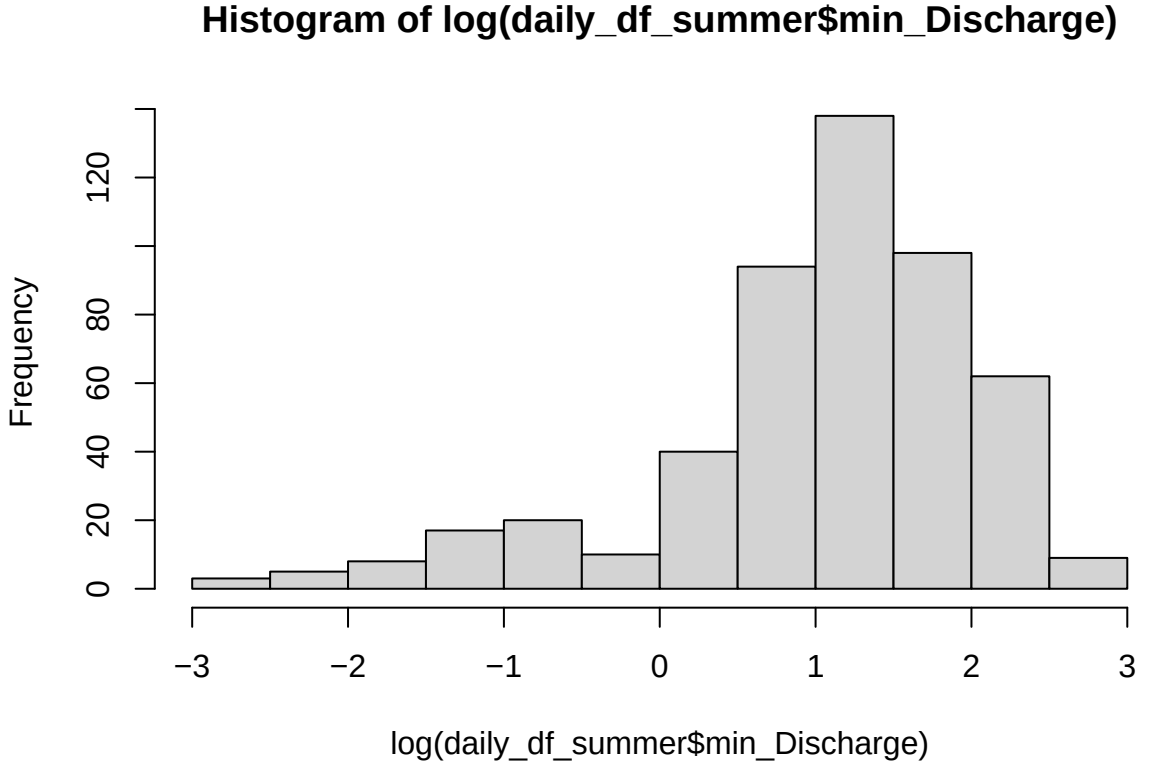



```
#stream T, Air T, DISCHARGE
```

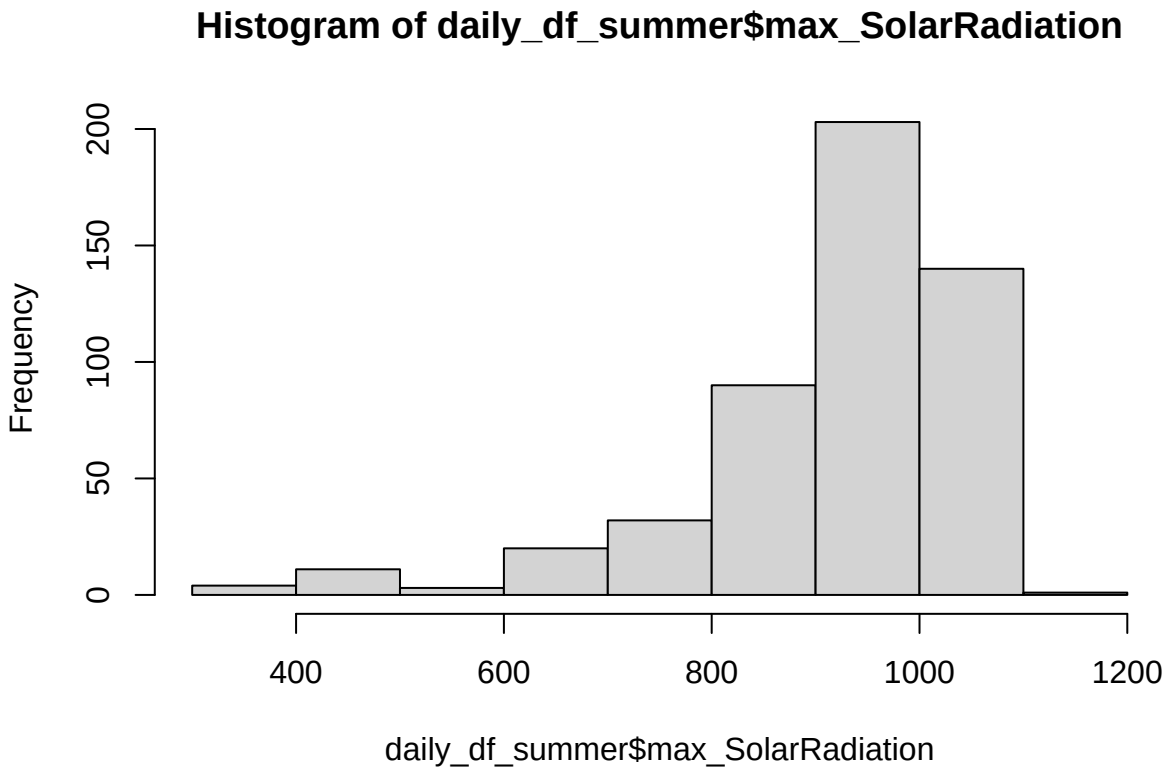
```
hist(daily_df_summer$max_StreamTemp)
```




```
hist(log(daily_df_summer$min_Discharge))
```




```
hist(daily_df_summer$max_SolarRadiation)
```




```
summary(lm(daily_df_summer$max_StreamTemp~ daily_df_summer$min_Discharge + daily_df_summer$max_AirTempe
```

```
##
## Call:
## lm(formula = daily_df_summer$max_StreamTemp ~ daily_df_summer$min_Discharge +
##     daily_df_summer$max_AirTemperature_C + daily_df_summer$max_SolarRadiation)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3833 -0.4612  0.1609  0.6096  1.8926
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         8.9741791  0.3176361  28.253   <2e-16 ***
## daily_df_summer$min_Discharge      -0.2449178  0.0141253 -17.339   <2e-16 ***
## daily_df_summer$max_AirTemperature_C  0.1765321  0.0090881  19.424   <2e-16 ***
## daily_df_summer$max_SolarRadiation    0.0002755  0.0003149   0.875    0.382
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8988 on 500 degrees of freedom
## Multiple R-squared:  0.688,  Adjusted R-squared:  0.6861
## F-statistic: 367.5 on 3 and 500 DF,  p-value: < 2.2e-16
```

```
summary(lm(daily_df_summer$max_StreamTemp~ log(daily_df_summer$min_Discharge) + daily_df_summer$max_Air
```

```
##
## Call:
## lm(formula = daily_df_summer$max_StreamTemp ~ log(daily_df_summer$min_Discharge) +
##     daily_df_summer$max_AirTemperature_C + daily_df_summer$max_SolarRadiation)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6199 -0.4033  0.1664  0.5968  1.7716
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         8.0501703  0.3039348  26.487   <2e-16 ***
## log(daily_df_summer$min_Discharge)  -0.6955668  0.0402610 -17.276   <2e-16 ***
## daily_df_summer$max_AirTemperature_C  0.1972112  0.0087520  22.533   <2e-16 ***
## daily_df_summer$max_SolarRadiation    0.0005020  0.0003164   1.587    0.113
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9 on 500 degrees of freedom
## Multiple R-squared:  0.6872, Adjusted R-squared:  0.6853
## F-statistic: 366.1 on 3 and 500 DF,  p-value: < 2.2e-16
```

```
summary(lm(daily_df_summer$max_StreamTemp~ log(daily_df_summer$min_Discharge) + daily_df_summer$max_Air
```

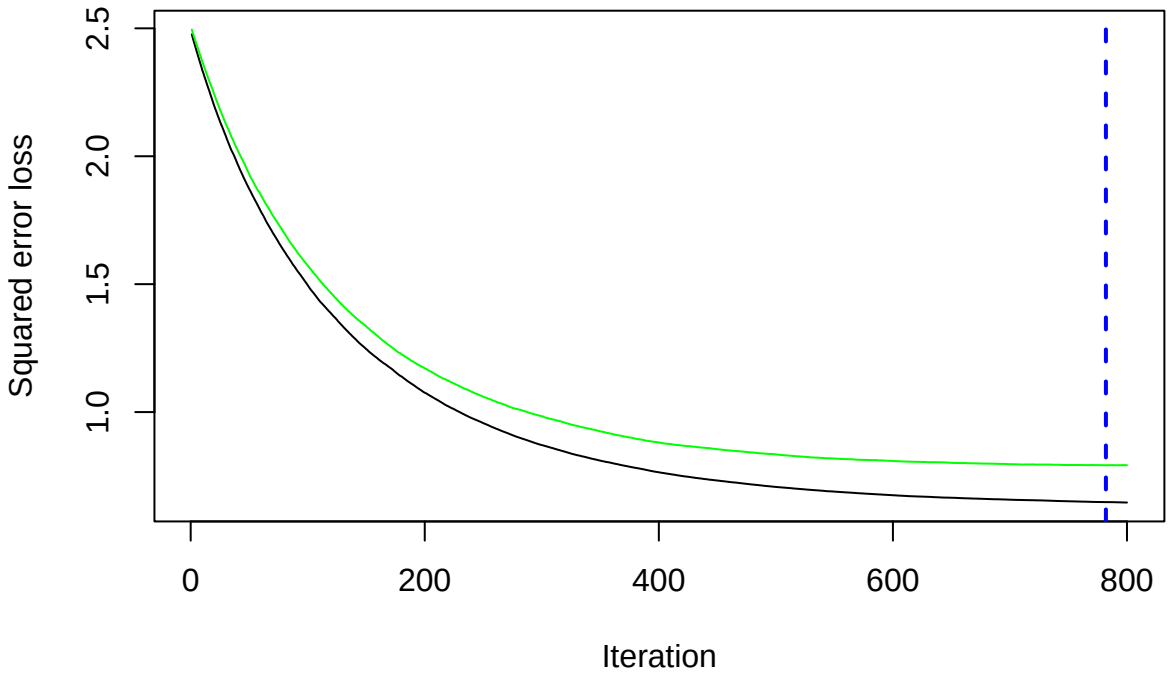
```
##
## Call:
## lm(formula = daily_df_summer$max_StreamTemp ~ log(daily_df_summer$min_Discharge) +
##     daily_df_summer$max_AirTemperature_C)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6108 -0.4395  0.1751  0.6079  1.7886
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         8.419310   0.195893   42.98   <2e-16 ***
## log(daily_df_summer$min_Discharge)  -0.689325   0.040129  -17.18   <2e-16 ***
## daily_df_summer$max_AirTemperature_C  0.201158   0.008404   23.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9013 on 501 degrees of freedom
## Multiple R-squared:  0.6856, Adjusted R-squared:  0.6843
## F-statistic: 546.2 on 2 and 501 DF,  p-value: < 2.2e-16
```

```
# set seed for generating random data.
set.seed(0)
# createDataPartition() function from the caret package to split the original dataset into a training a
variables<-c("max_StreamTemp","min_Discharge","max_AirTemperature_C", "max_SolarRadiation")
parts = createDataPartition(  daily_df_summer$max_StreamTemp , p = .8, list = F)
train = daily_df_summer[parts, variables ]
test =  daily_df_summer[-parts, variables  ]
# feature and target array
test_x = test[, -1]
test_y = test[, 1]
```

```
model_gbm = gbm(train$max_StreamTemp      ~.,
                data = train,
                distribution = "gaussian",
```

```
                cv.folds = 10,
                shrinkage = .01,
                n.minobsinnode = 10,
                n.trees = 800)

# model performance
perf_gbm1 = gbm.perf( model_gbm, method = "cv")
```



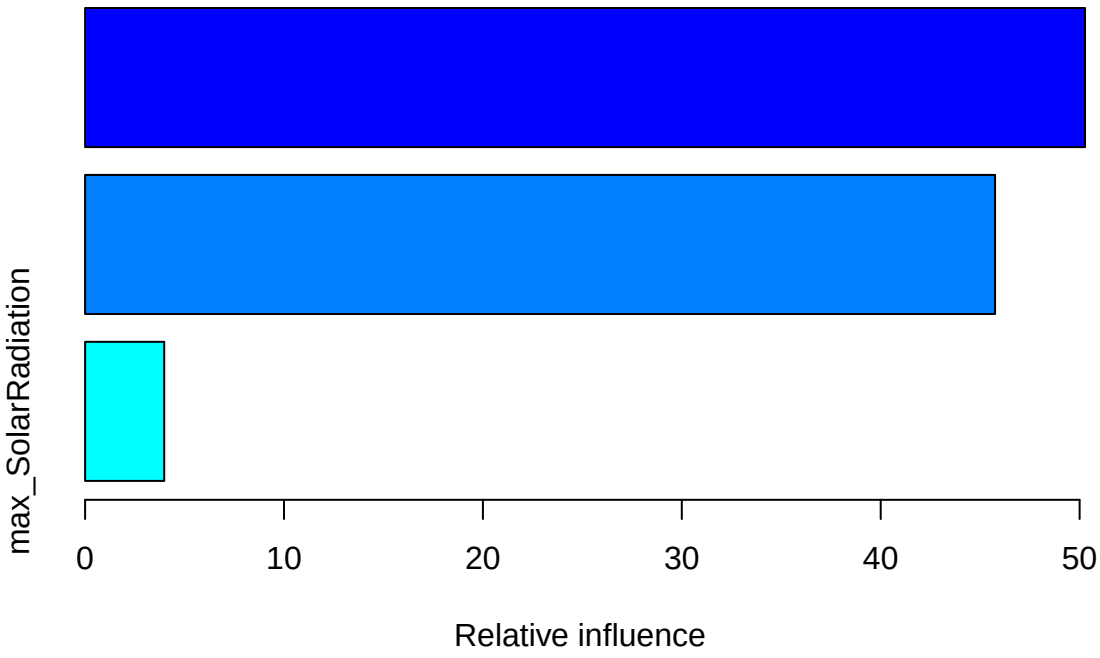

```
print(model_gbm)
```

```
## gbm(formula = train$max_StreamTemp ~ ., distribution = "gaussian",
##     data = train, n.trees = 800, n.minobsinnode = 10, shrinkage = 0.01,
##     cv.folds = 10)
## A gradient boosted model with gaussian loss function.
## 800 iterations were performed.
## The best cross-validation iteration was 782.
## There were 3 predictors of which 3 had non-zero influence.
```

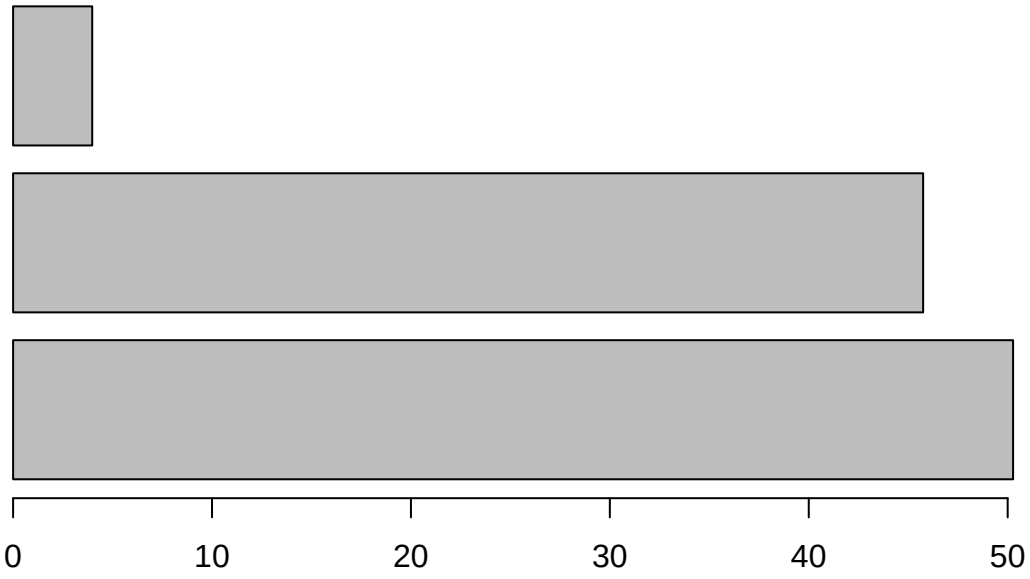
```
summary(model_gbm)
```

```
##                                     var   rel.inf
## max_AirTemperature_C max_AirTemperature_C 50.271972
## min_Discharge               min_Discharge 45.749851
## max_SolarRadiation     max_SolarRadiation  3.978177
```

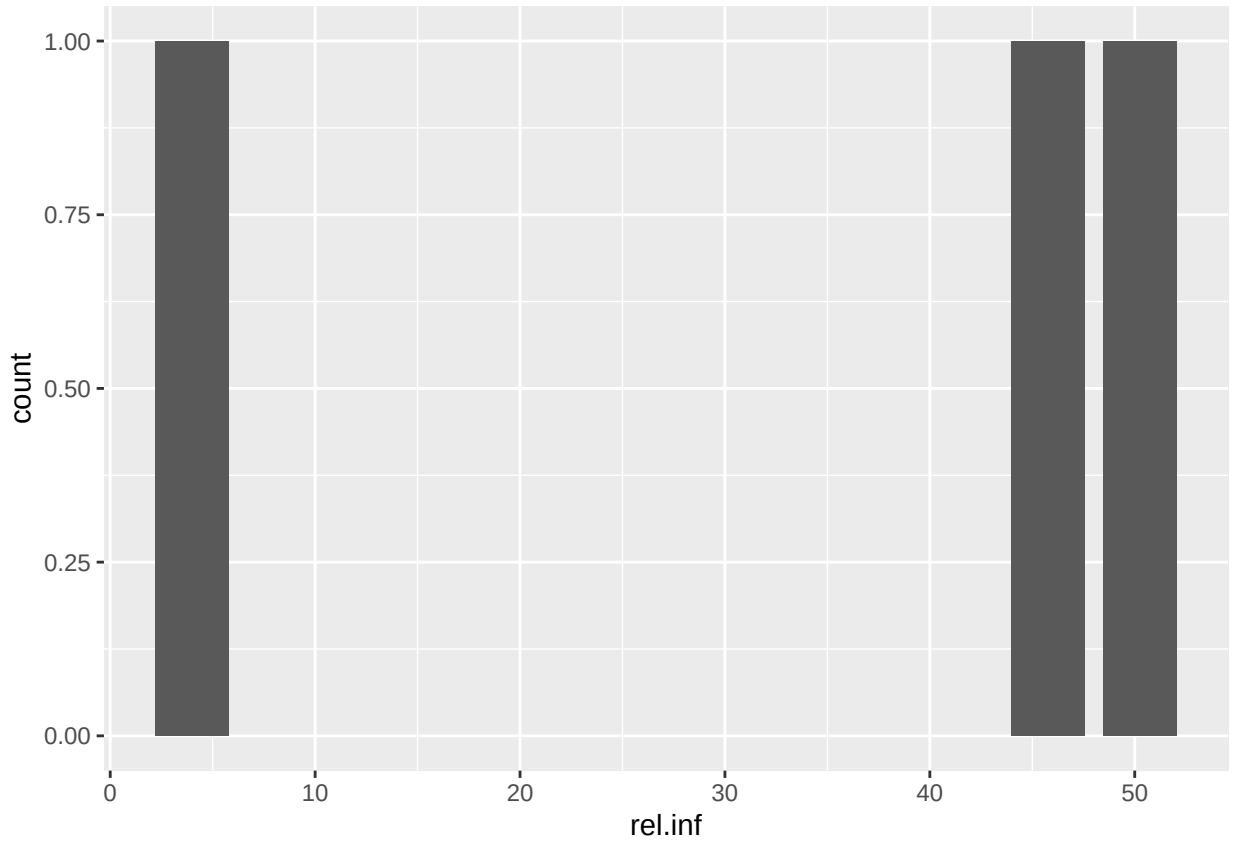
```
rinf<-summary(model_gbm)
```



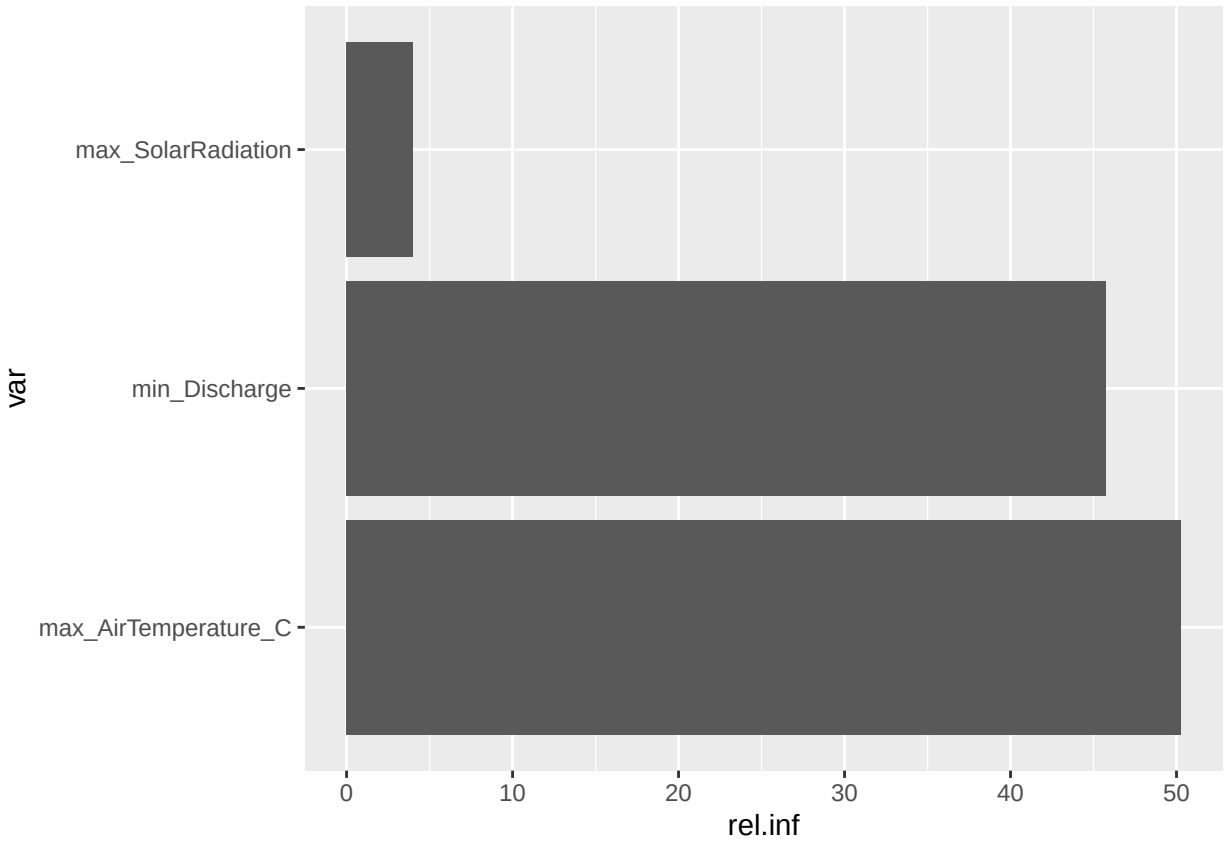

```
barplot( rinf$rel.inf , horiz = TRUE, las = 1)
```




```
ggplot(rinf, aes(rel.inf)) + geom_bar()
```

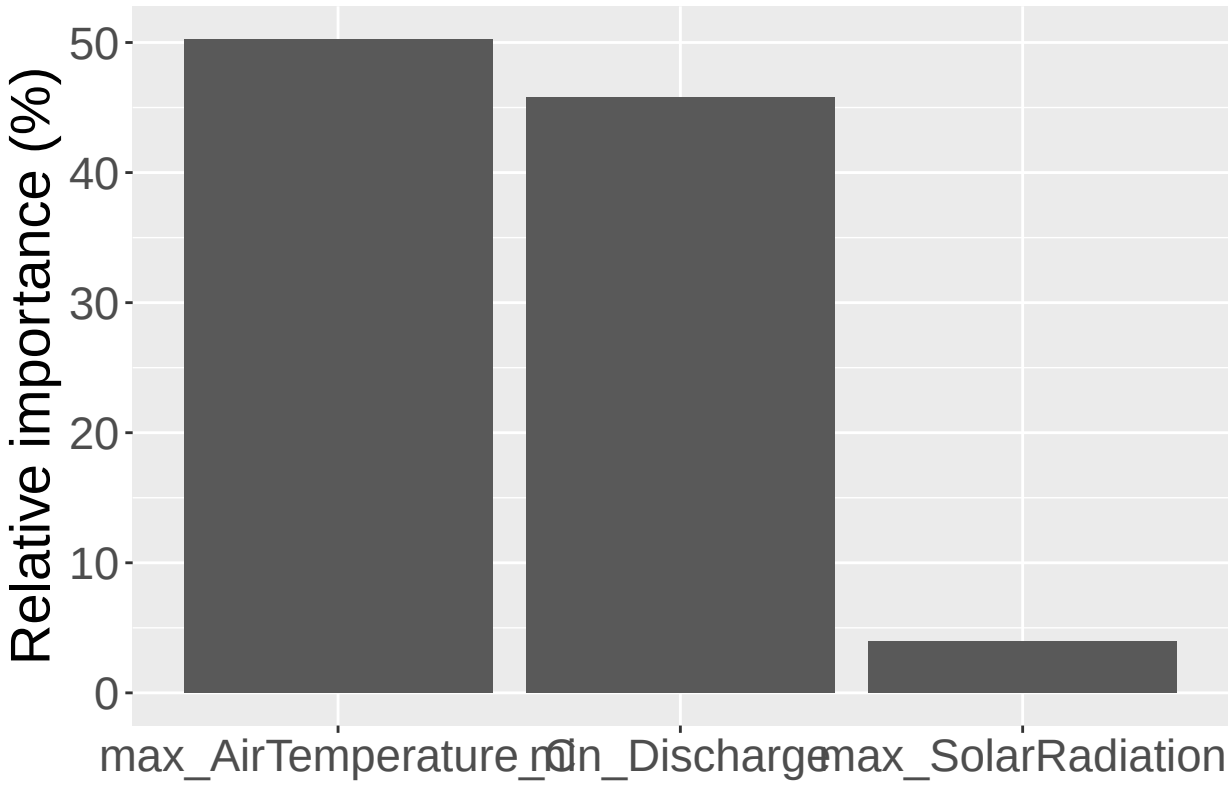



```
rinf$var<- factor(rinf$var, levels=c(  "max_AirTemperature_C" ,"min_Discharge"   ,      "max_SolarRadiat
ggplot( rinf, aes( var , rel.inf ))+  geom_col()+
  coord_flip()
```




```
ggplot( rinf )+  geom_bar(  aes( x=var, y= rel.inf),  stat = "summary")+ scale_x_discrete(labels=  c( "
```

```
## No summary function supplied, defaulting to `mean_se()`
```




```
pred_y = predict.gbm(model_gbm, test_x)
```

```
## Using 782 trees...
```

```
residuals =  test_y$max_StreamTemp  - pred_y
summary(test_y$max_StreamTemp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.30   11.25   12.20   11.96   13.10   14.90
```

```
xlim=c(5,20)
RMSE = sqrt(mean(residuals^2))
cat('The root mean square error of the test data is ', round(RMSE,3),'\n')
```

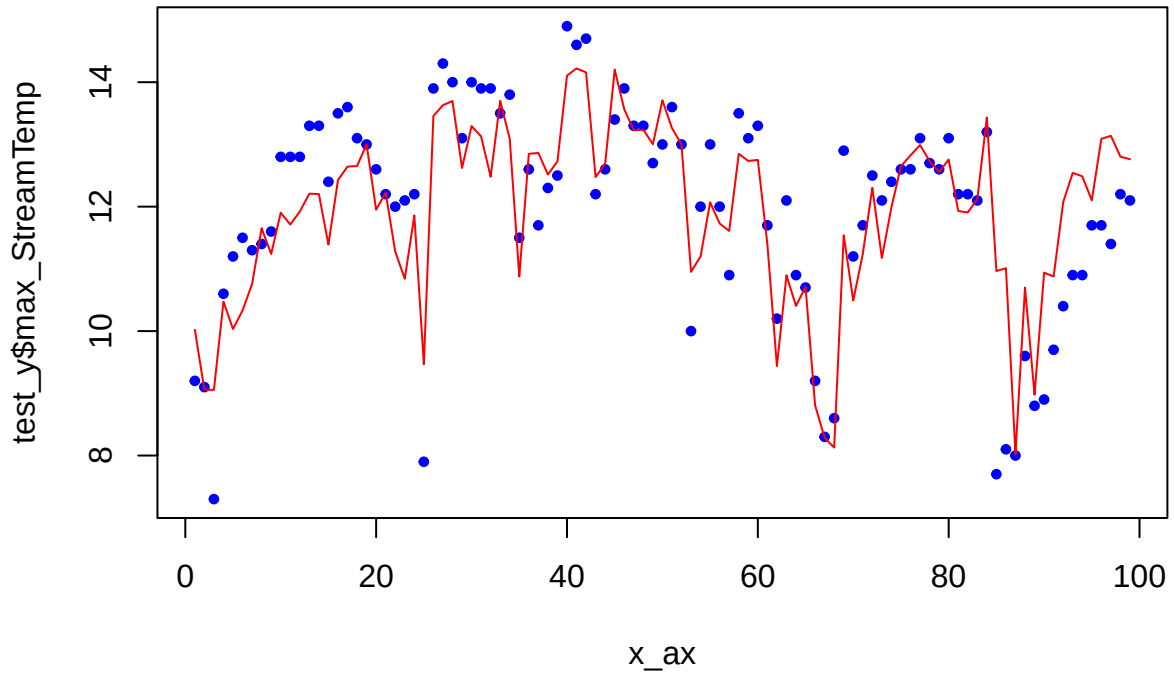
```
## The root mean square error of the test data is  0.9
```

```
y_test_mean = mean( test_y$max_StreamTemp )
# Calculate total sum of squares
tss =  sum(( test_y$max_StreamTemp  - y_test_mean)^2 )
# Calculate residual sum of squares
rss =  sum(residuals^2)
# Calculate R-squared
rsq  =  1 - (rss/tss)
cat('The R-square of the test data is ', round(rsq,3), '\n')
```

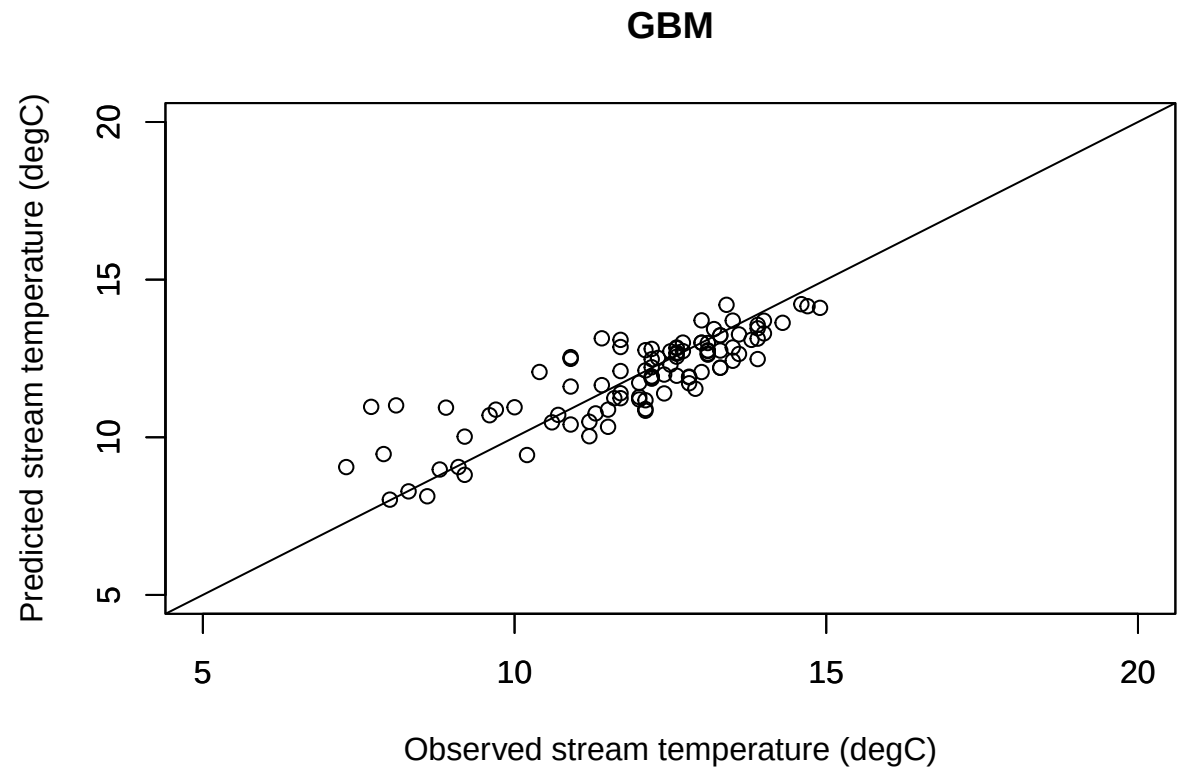
```
## The R-square of the test data is  0.719
```

```
# visualize the model, actual and predicted data
x_ax = 1:length(pred_y)
plot(x_ax, test_y$max_StreamTemp , col="blue", pch=20, cex=.9)
```

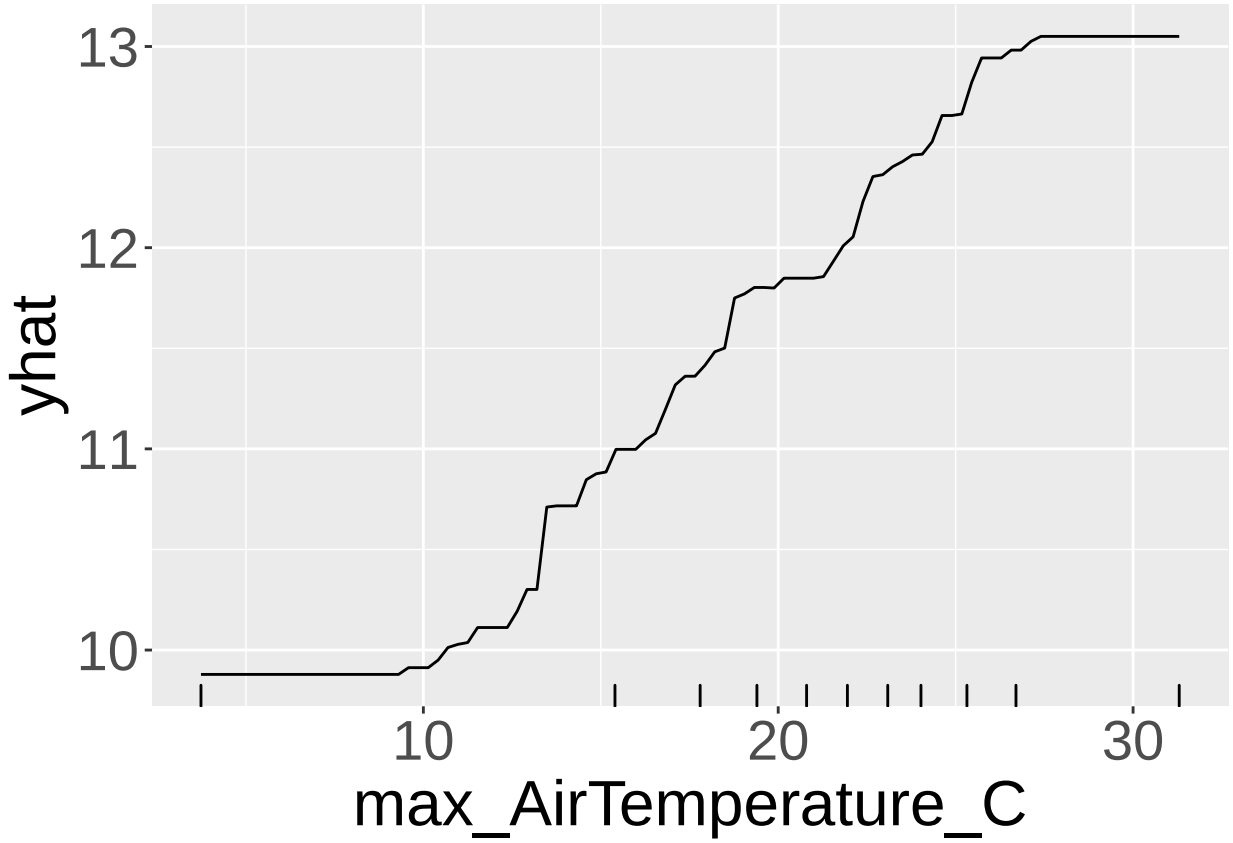
```
lines(x_ax, pred_y, col="red", pch=20, cex=.9)
```




```
plot(  test_y$max_StreamTemp, pred_y,xlim= xlim ,ylim= xlim, xlab="Observed stream temperature (degC)",
par(new=T)
x=seq(1,30)
plot(x,x,type="l",xlim= xlim ,ylim= xlim,xlab="",ylab="")
```

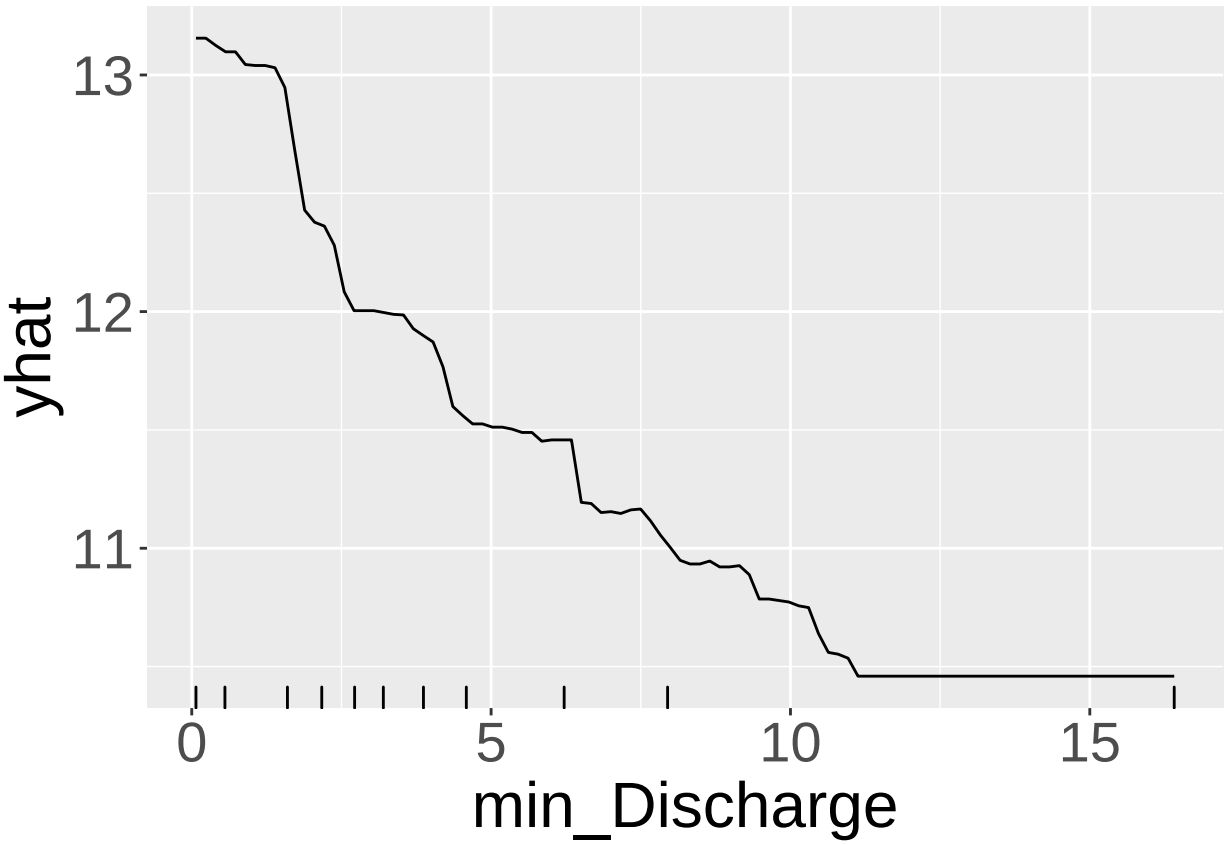


```
model_gbm %>%
  partial(pred.var =  "max_AirTemperature_C"    , n.trees = model_gbm$n.trees, grid.resolution = 100) %>%
  autoplot(rug = TRUE, train = train)+theme(axis.text=element_text(size=21),
        axis.title=element_text(size=24))
```




```
#,"min_Discharge"

model_gbm %>%
  partial(pred.var =  "min_Discharge"    , n.trees = model_gbm$n.trees, grid.resolution = 100) %>%
  autoplot(rug = TRUE, train = train)+theme(axis.text=element_text(size=21),
        axis.title=element_text(size=24))
```


yhat
13
12
11
0
5
10
15
min_Discharge