```r
# Import library
library(readxl)
library(dplyr)
```

```
## Warning: 套件 'dplyr' 是用 R 版本 4.3.2 來建造的
```

```
##
## 載入套件：'dplyr'
```

```
## 下列物件被遮斷自 'package:stats':
##
##     filter, lag
```

```
## 下列物件被遮斷自 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
```

```
## Warning: 套件 'ggplot2' 是用 R 版本 4.3.3 來建造的
```

```r
library(cowplot)
```

```
## Warning: 套件 'cowplot' 是用 R 版本 4.3.3 來建造的
```

```r
# Random forest library
library(randomForest)
```

```
## randomForest 4.7-1.1
```

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

```
##
## 載入套件：'randomForest'
```

```
## 下列物件被遮斷自 'package:ggplot2':
##
##     margin
```

```
## 下列物件被遮斷自 'package:dplyr':
##
##     combine
```

```r
library(psych)
```

```
## Warning: 套件 'psych' 是用 R 版本 4.3.3 來建造的
```

```
##
## 載入套件：'psych'
```

```
## 下列物件被遮斷自 'package:randomForest':
##
##     outlier
```

```
## 下列物件被遮斷自 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(keras)
# Lasso
library(glmnet)
```

```
## 載入需要的套件：Matrix
```

```
## Warning: 套件 'Matrix' 是用 R 版本 4.3.3 來建造的
```

```
## Loaded glmnet 4.1-8
```

```r
library(caret)
```

```
## 載入需要的套件：lattice
```

```
## Warning: 套件 'lattice' 是用 R 版本 4.3.2 來建造的
```

```r
library(boot)
```

```
## Warning: 套件 'boot' 是用 R 版本 4.3.3 來建造的
```

```
##
## 載入套件：'boot'
```

```
## 下列物件被遮斷自 'package:lattice':
##
##     melanoma
```

```
## 下列物件被遮斷自 'package:psych':
##
##     logit
```

```r
library(car)
```

```
## 載入需要的套件：carData
```

```
##
## 載入套件：'car'
```

```
## 下列物件被遮斷自 'package:boot':
##
##     logit
```

```
## 下列物件被遮斷自 'package:psych':
##
##     logit
```

```
## 下列物件被遮斷自 'package:dplyr':
##
##     recode
```

```r
library(scales)
```

```
## Warning: 套件 'scales' 是用 R 版本 4.3.2 來建造的
```

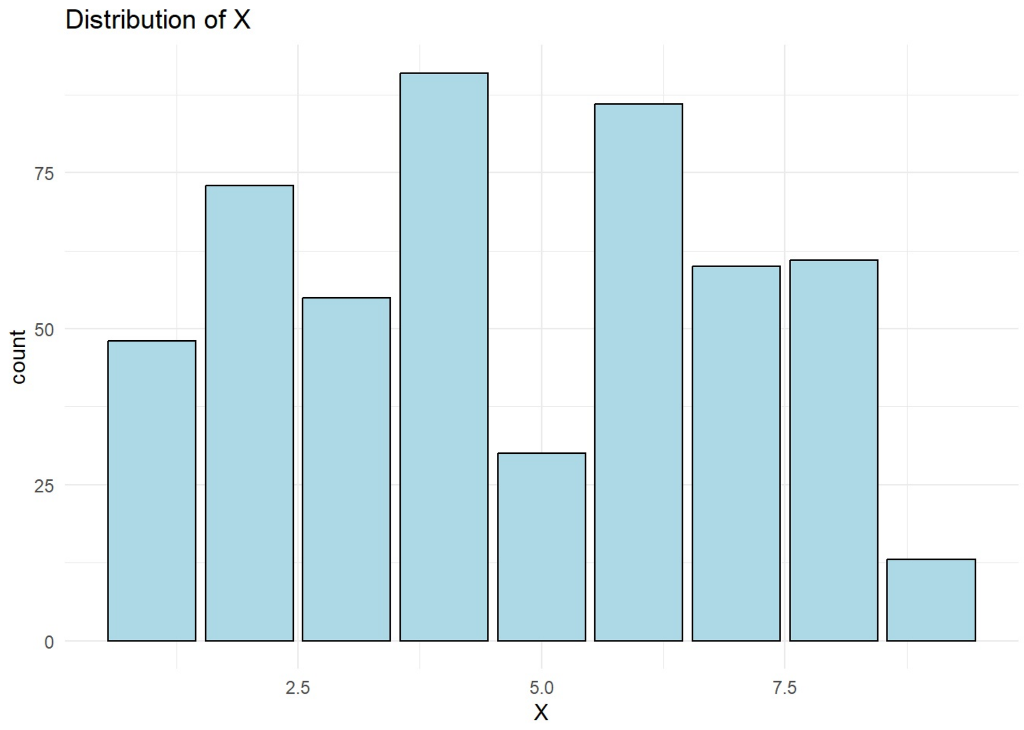
```
##
## 載入套件：'scales'
```

```
## 下列物件被遮斷自 'package:psych':
##
##     alpha, rescale
```

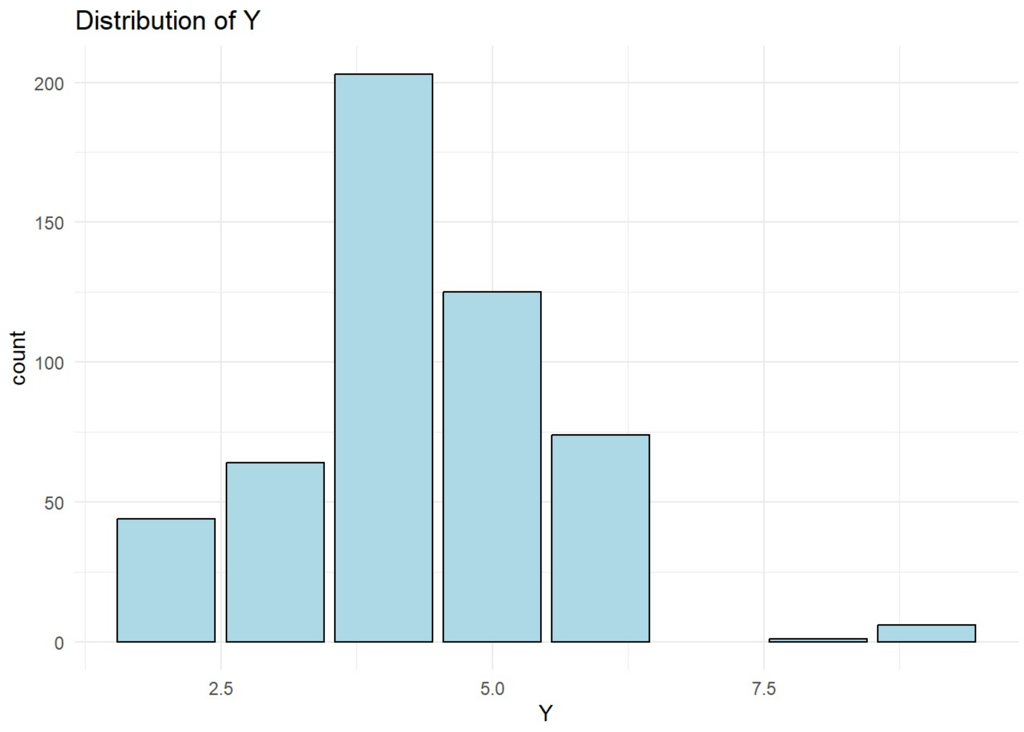
```r
library(glm2)
```

```
#import data
forestfires <- read.csv("C:\\Users\\54088\\OneDrive\\桌面\\SW\\S5 2023\\STAT 4893W\\Second Project\\forestfires.cs
v")

# Explaination of the covariates
#   1. X - x-axis spatial coordinate within the Montesinho park map: 1 to 9
#   2. Y - y-axis spatial coordinate within the Montesinho park map: 2 to 9

#   3. month - month of the year: 'jan' to 'dec'
#   4. day - day of the week: 'mon' to 'sun'

#   5. FFMC - Fine Fuel Moisture Code index from the FWI system: 18.7 to 96.20
#   6. DMC - Duff Moisture index from the FWI system: 1.1 to 291.3
#   7. DC - Drought Code index from the FWI system: 7.9 to 860.6

#   8. ISI - Initial Spread Index from the FWI system: 0.0 to 56.10

#   9. temp - Temperature in Celsius degrees: 2.2 to 33.30
#   10. RH - Relative Humidity in %: 15.0 to 100
#   11. wind - Wind Speed in km/h: 0.40 to 9.40
#   12. rain - outside rain in mm/m2 : 0.0 to 6.4
#   13. area - the burned area of the forest (in ha): 0.00 to 1090.84
#   (this output variable is very skewed towards 0.0, thus it may make
#   sense to model with the logarithm transform).

# Rearrange the variables
forestfires <- forestfires %>%
  mutate(month = case_when(month %in% "jan" ~ 1,
                           month %in% "feb" ~ 2,
                           month %in% "mar" ~ 3,
                           month %in% "apr" ~ 4,
                           month %in% "may" ~ 5,
                           month %in% "jun" ~ 6,
                           month %in% "jul" ~ 7,
                           month %in% "aug" ~ 8,
                           month %in% "sep" ~ 9,
                           month %in% "oct" ~ 10,
                           month %in% "nov" ~ 11,
                           month %in% "dec" ~ 12))

forestfires <- forestfires %>%
  mutate(day = case_when(day %in% "mon" ~ 1,
                         day %in% "tue" ~ 2,
                         day %in% "wed" ~ 3,
                         day %in% "thu" ~ 4,
                         day %in% "fri" ~ 5,
                         day %in% "sat" ~ 6,
                         day %in% "sun" ~ 7))

forestfires$RH <- as.numeric(forestfires$RH)
```

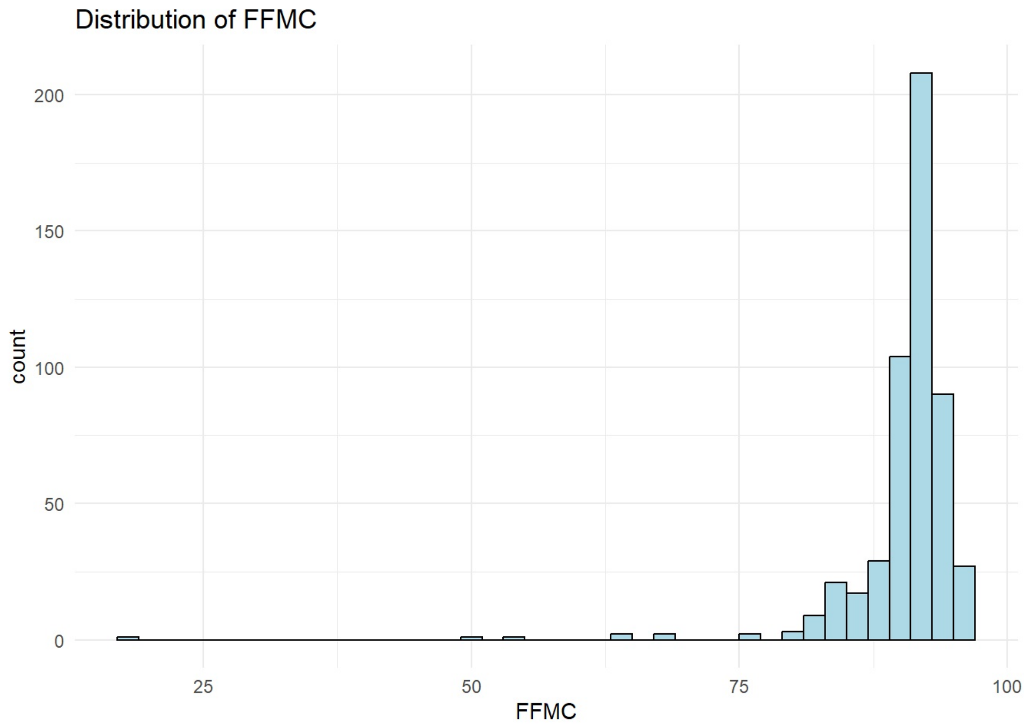
```
EDA_forest <- forestfires
par(mfrow = c(1, 9))

# X
# Plot for X
ggplot(EDA_forest, aes(x = X)) +
  geom_bar(fill = "lightblue", color = "black") +
  theme_minimal() +
  ggtitle("Distribution of X")
```



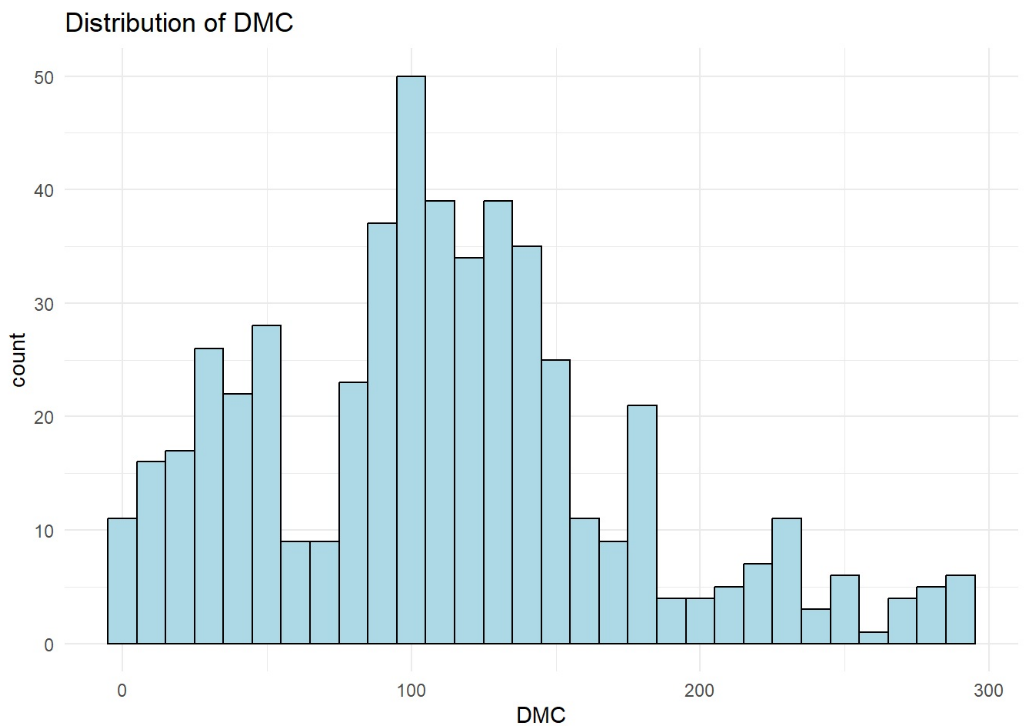

```
# Y
# Plot for Y
ggplot(EDA_forest, aes(x = Y)) +
  geom_bar(fill = "lightblue", color = "black") +
  theme_minimal() +
  ggtitle("Distribution of Y")
```



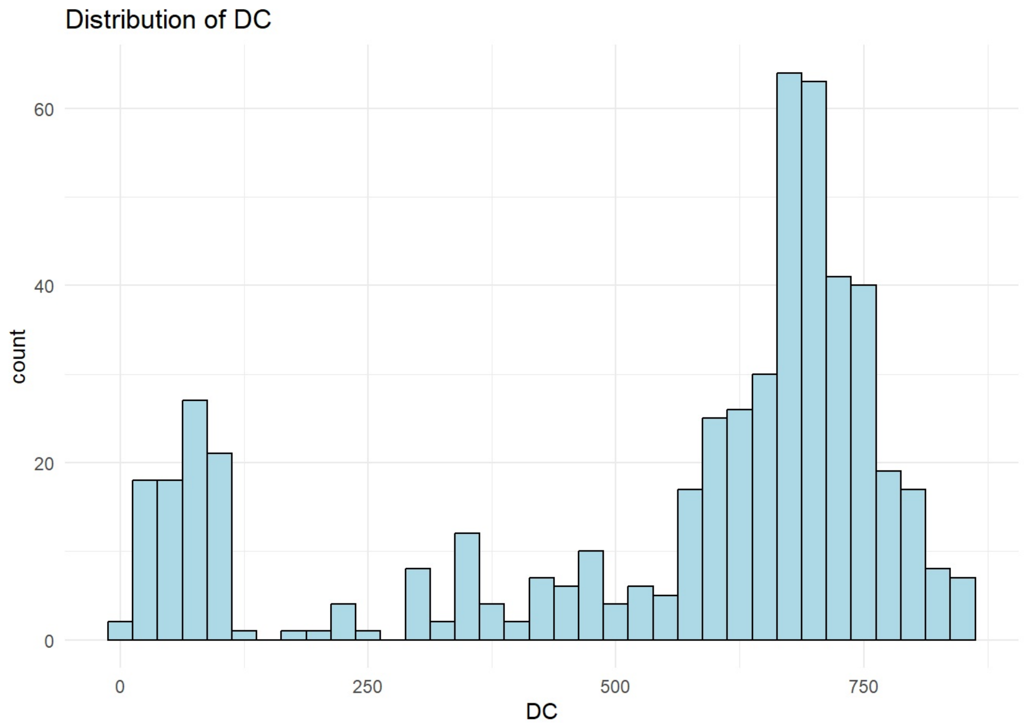

```
# FFMC
ggplot(EDA_forest, aes(x = FFMC)) +
  geom_histogram(binwidth = 2, fill = "lightblue", color = "black") +
  theme_minimal() +
  ggtitle("Distribution of FFMC")
```



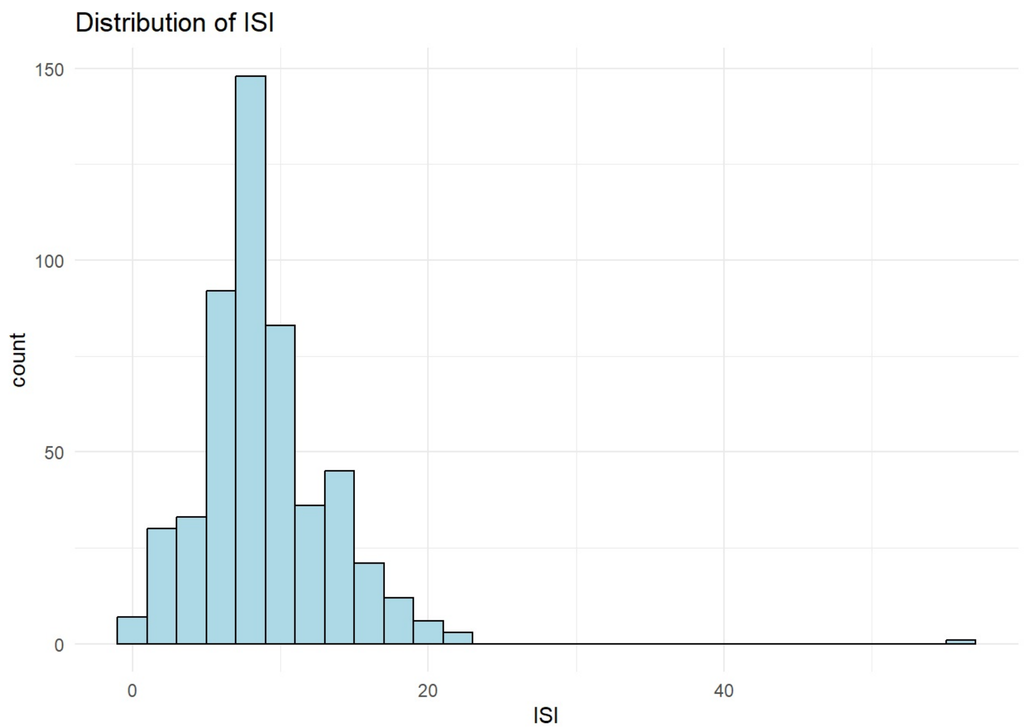

```r
# DMC
ggplot(EDA_forest, aes(x = DMC)) +
  geom_histogram(binwidth = 10, fill = "lightblue", color = "black") +
  theme_minimal() +
  ggtitle("Distribution of DMC")
```



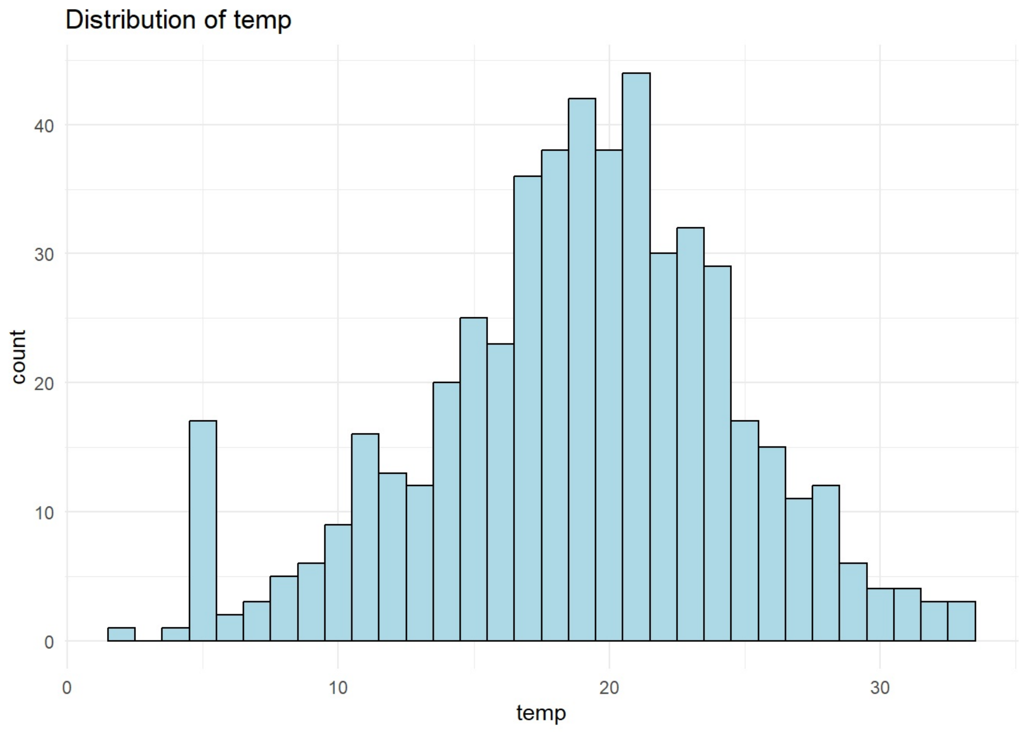

```r
# DC
ggplot(EDA_forest, aes(x = DC)) +
  geom_histogram(binwidth = 25, fill = "lightblue", color = "black") +
  theme_minimal() +
  ggtitle("Distribution of DC")
```



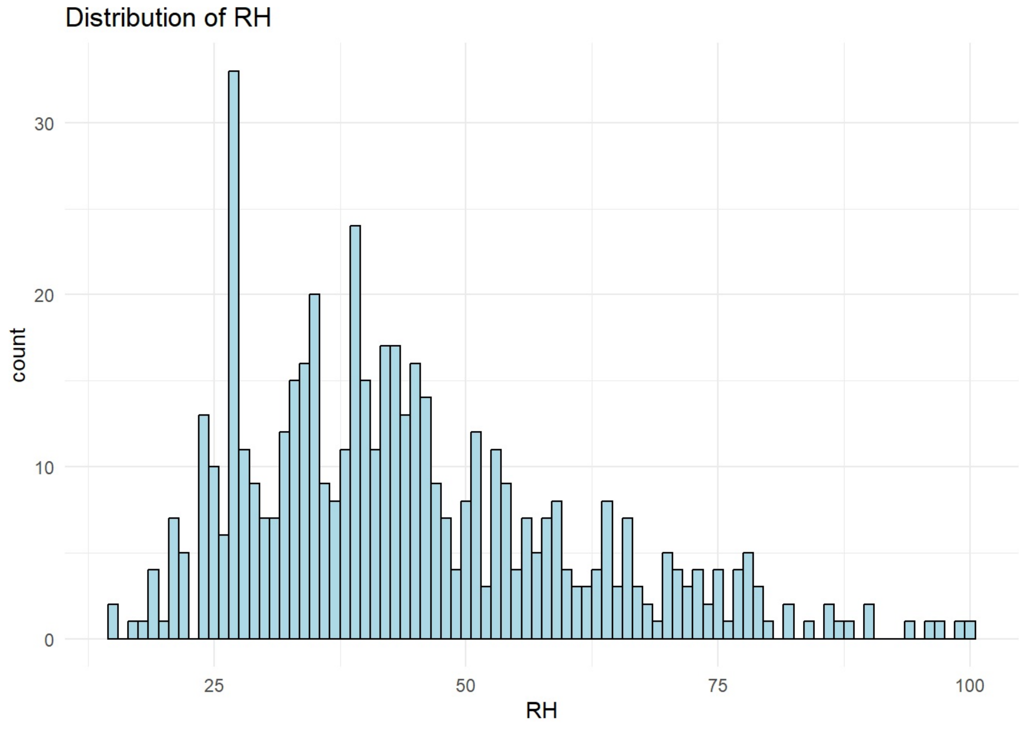

```r
# ISI
ggplot(EDA_forest, aes(x = ISI)) +
  geom_histogram(binwidth = 2, fill = "lightblue", color = "black") +
  theme_minimal() +
  ggtitle("Distribution of ISI")
```



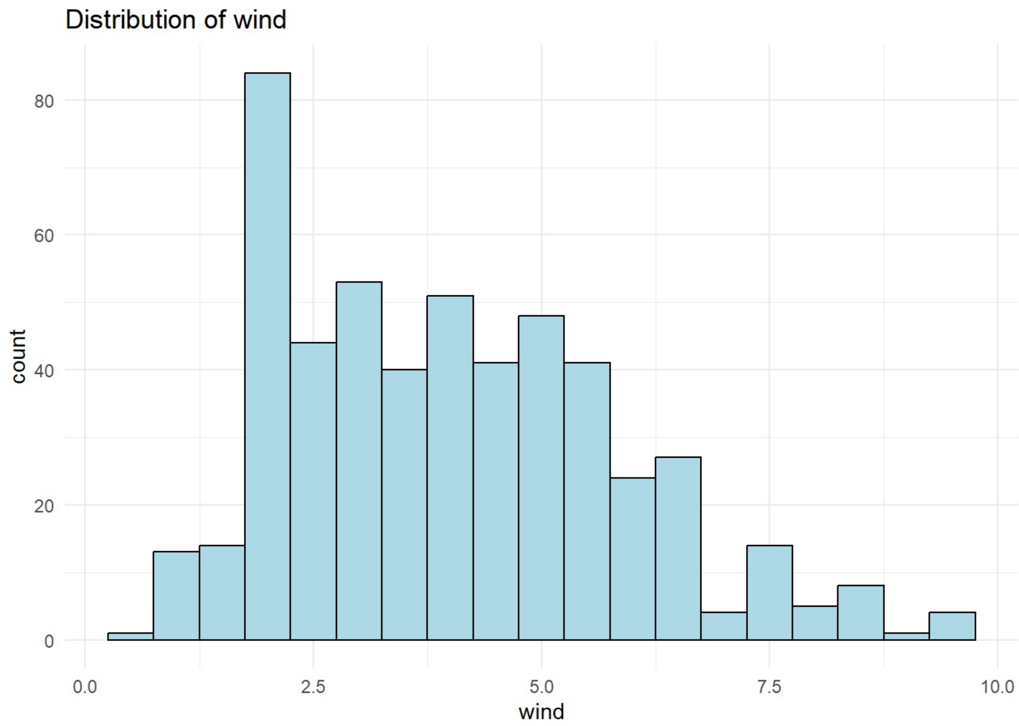

```r
# temp
ggplot(EDA_forest, aes(x = temp)) +
  geom_histogram(binwidth = 1, fill = "lightblue", color = "black") +
  theme_minimal() +
  ggtitle("Distribution of temp")
```



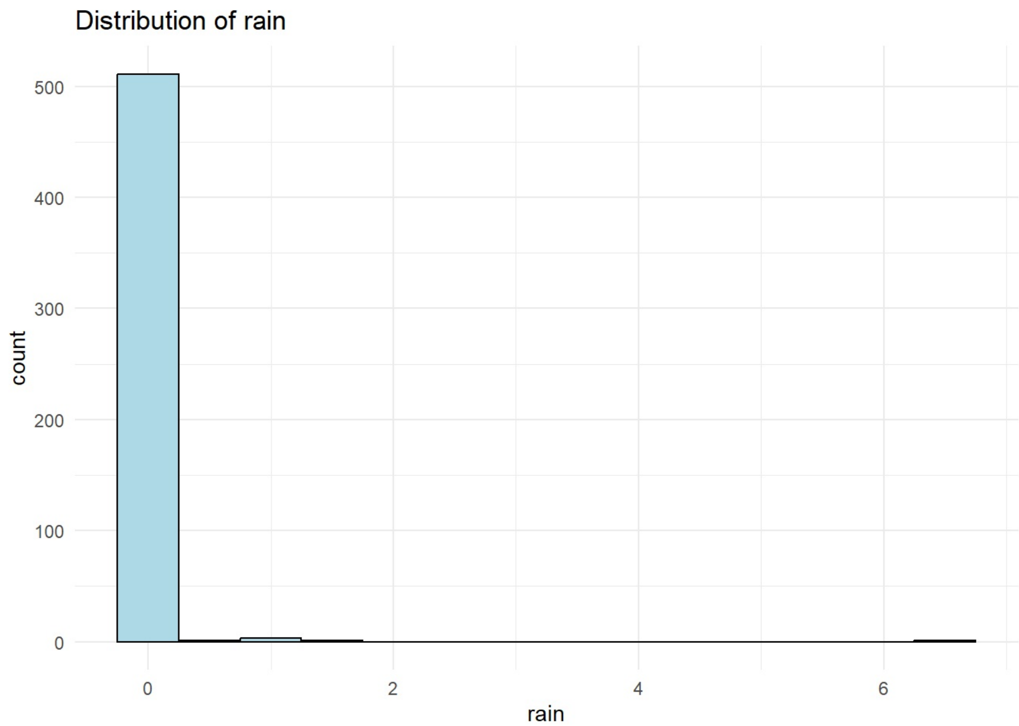

```r
# RH
ggplot(EDA_forest, aes(x = RH)) +
  geom_histogram(binwidth = 1, fill = "lightblue", color = "black") +
  theme_minimal() +
  ggtitle("Distribution of RH")
```



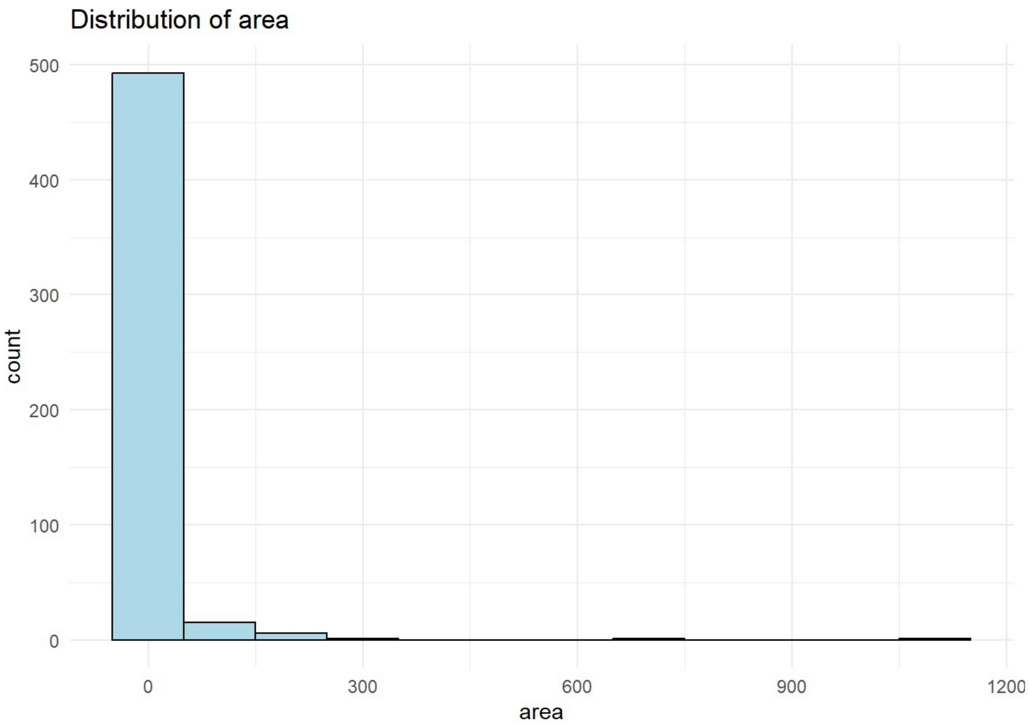

```r
# wind
ggplot(EDA_forest, aes(x = wind)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
  theme_minimal() +
  ggtitle("Distribution of wind")
```



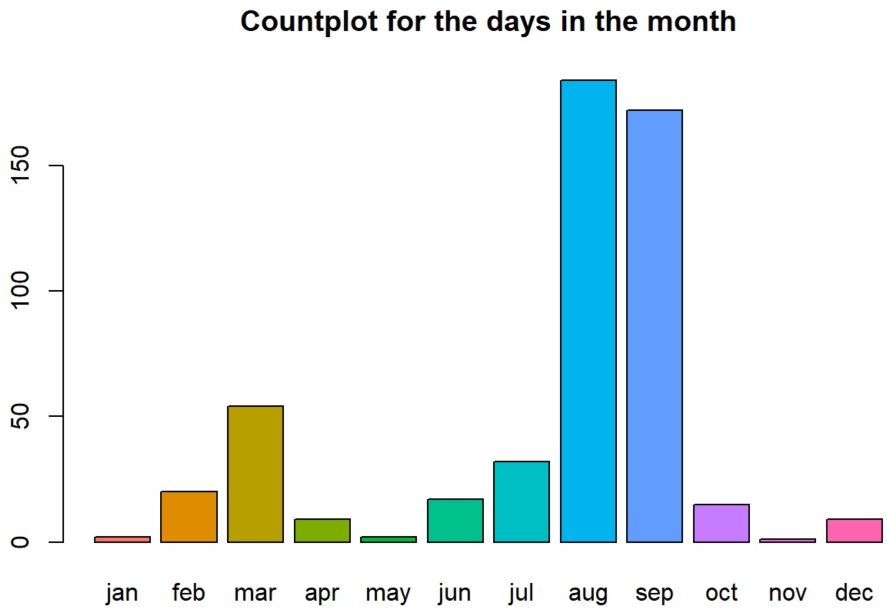

```
# rain
ggplot(EDA_forest, aes(x = rain)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
  theme_minimal() +
  ggtitle("Distribution of rain")
```



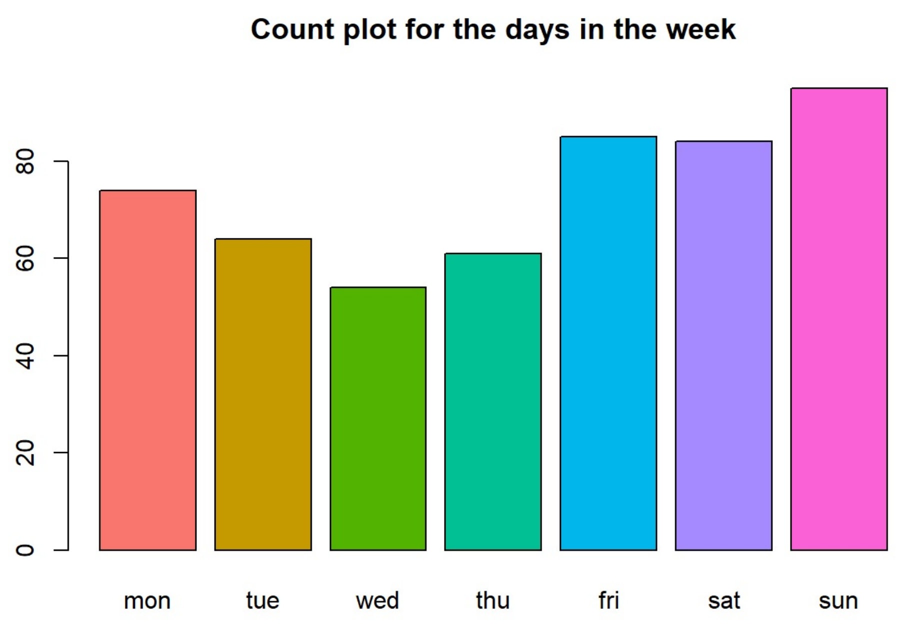

```
# area
ggplot(EDA_forest, aes(x = area)) +
  geom_histogram(binwidth = 100, fill = "lightblue", color = "black") +
  theme_minimal() +
  ggtitle("Distribution of area")
```




```
# EDA for month
month1 <- sum(forestfires$month == 1)
month2 <- sum(forestfires$month == 2)
month3 <- sum(forestfires$month == 3)
month4 <- sum(forestfires$month == 4)
month5 <- sum(forestfires$month == 5)
month6 <- sum(forestfires$month == 6)
month7 <- sum(forestfires$month == 7)
month8 <- sum(forestfires$month == 8)
month9 <- sum(forestfires$month == 9)
month10 <- sum(forestfires$month == 10)
month11 <- sum(forestfires$month == 11)
month12 <- sum(forestfires$month == 12)

colors <- c("#F8766D", "#DE8C00", "#B79F00", "#7CAE00", "#00BA38", "#00C08B", "#00BFC4","#00B4F0", "#619CFF", "#C
77CFF", "#F564E3", "#FF64B0")
barplot(table(forestfires$month), col = colors, main = "Countplot for the days in the month", names.arg = c("jan"
, "feb", "mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", "dec"))
```




```
colors <- c("#F8766D", "#C49A00", "#53B400", "#00C094", "#00B6EB", "#A58AFF", "#FB61D7")
barplot(table(forestfires$day), col = colors, main = "Count plot for the days in the week", names.arg = c("mon",
"tue", "wed", "thu", "fri", "sat", "sun"))
```




```r
# Zero-inflated poisson model
zim_fires <- forestfires
zim_fires$area <- round(zim_fires$area)

library(pscl)
```

```
## Warning: 套件 'pscl' 是用 R 版本 4.3.3 來建造的
```

```
## Classes and Methods for R originally developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University (2002-2015),
## by and under the direction of Simon Jackman.
## hurdle and zeroinfl functions by Achim Zeileis.
```

```r
set.seed(4893)
nr = nrow(zim_fires)
train_indices = sample(nr, nr*0.7)
train_zim <- zim_fires[train_indices, ]
test_zim <- zim_fires[-train_indices, ]

# The former part specifies the count model formula implies the predictive variable affect the count model and th
e latter part after | is the zero-inflated model, implies what predictive variable affect the probability of gett
ing a zero.
model_zim <- zeroinfl(area ~ .|., data = train_zim, family = "negbin")
```

```
## Warning in optim(fn = countloglikfun, gr = countgradfun, par = c(lmstart, :
## 控制裡不明的名稱 family
```

```
## Warning in optim(fn = loglikfun, gr = gradfun, par = c(start$count, start$zero,
## : 控制裡不明的名稱 family
```

```r
summary(model_zim)
```

```
##
## Call:
## zeroinfl(formula = area ~ . | ., data = train_zim, family = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.3745 -0.9171 -0.7740 -0.3366 36.7033
##
## Count model coefficients (poisson with log link):
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.1152547  0.8449266  -7.238 4.57e-13 ***
## X            0.1680060  0.0076128  22.069  < 2e-16 ***
## Y           -0.0761806  0.0140181  -5.434 5.50e-08 ***
## month        0.3468054  0.0168799  20.545  < 2e-16 ***
## day          0.1676006  0.0082338  20.355  < 2e-16 ***
## FFMC         0.0775850  0.0094607   8.201 2.39e-16 ***
## DMC          0.0070161  0.0003697  18.979  < 2e-16 ***
## DC          -0.0027303  0.0001782 -15.321  < 2e-16 ***
## ISI         -0.0594745  0.0066032  -9.007  < 2e-16 ***
## temp         0.0388473  0.0042675   9.103  < 2e-16 ***
## RH          -0.0325124  0.0015430 -21.072  < 2e-16 ***
## wind         0.0298290  0.0103863   2.872  0.00408 **
## rain        -1.3261287  0.5151172  -2.574  0.01004 *
##
## Zero-inflation model coefficients (binomial with logit link):
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.4109482  4.7552237  -0.507   0.6121
## X           -0.0256428  0.0556096  -0.461   0.6447
## Y           -0.0482048  0.1065380  -0.452   0.6509
## month       -0.0783630  0.1182980  -0.662   0.5077
## day         -0.0393755  0.0535141  -0.736   0.4619
## FFMC         0.0501860  0.0543334   0.924   0.3557
## DMC         -0.0003481  0.0026734  -0.130   0.8964
## DC           0.0004185  0.0013652   0.307   0.7592
## ISI         -0.0040021  0.0314903  -0.127   0.8989
## temp        -0.0310949  0.0314498  -0.989   0.3228
## RH          -0.0015408  0.0097342  -0.158   0.8742
## wind        -0.1264051  0.0670626  -1.885   0.0594 .
## rain        -1.0178887  2.1121367  -0.482   0.6299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 36
## Log-likelihood: -6758 on 26 Df
```

```
# Large pearson residual indicates poor fit or outliers.
# For the non-zero counts of the area variable. Every predictive variables have strong association with the area.
But, when it comes to the probability of the zeros, the increase of variables "day", "DC", "ISI", the probability
of area being zero increase.

#(poisson with log link): meaning the count part is modeled using a Poisson distribution. log link helps with mak
ing sure that the result is positive integer.

#(binomial with logit link): Binomial (0 or 1). The logit translates the linear combination of predictors into a
probability between 0 and 1.

# The log likelihood shows how good the model fits the data, the higher the number is, the greater the model fits
the data.

predict_zim <- predict(model_zim, newdata = test_zim, type = "count")

#MAE
mae_zim <- mean(abs(predict_zim - test_zim$area))
mae_zim
```

```
## [1] 25.9363
```

```
#MAD
mad_zim <- median(abs(predict_zim - mean(test_zim$area)))
mad_zim
```

```
## [1] 10.61392
```

```r
# Accurancy
actual_zero <- ifelse(test_zim$area == 0, 1, 0)
predicted <- ifelse(predict_zim > 0.5, 1, 0)
table(actual_zero, predicted)
```

```
##            predicted
## actual_zero  0  1
##           0  1 88
##           1  2 65
```

```r
mean(predicted!=test_zim$area)
```

```
## [1] 0.9423077
```

```r
# hurdle model
hurdle_fires <- forestfires
hurdle_fires$area <- round(hurdle_fires$area)

set.seed(4893)
nr = nrow(hurdle_fires)
train_indices = sample(nr, nr*0.7)
train_hurdle <- hurdle_fires[train_indices, ]
test_hurdle <- hurdle_fires[-train_indices, ]

model_hurdle <- hurdle(area ~ .|., data = train_hurdle, family = "negbin")
```

```
## Warning in optim(fn = countDist, gr = countGrad, par = c(start$count, if (dist
## == : 控制裡不明的名稱 family
```

```
## Warning in optim(fn = zeroDist, gr = zeroGrad, par = c(start$zero, if
## (zero.dist == : 控制裡不明的名稱 family
```

```r
summary(model_hurdle)
```

```
##
## Call:
## hurdle(formula = area ~ . | ., data = train_hurdle, family = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.3763 -0.9120 -0.7759 -0.3408 37.6062
##
## Count model coefficients (truncated poisson with log link):
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.1591822  0.8666028  -7.107 1.18e-12 ***
## X            0.1683099  0.0076351  22.044  < 2e-16 ***
## Y           -0.0762455  0.0140340  -5.433 5.54e-08 ***
## month        0.3484148  0.0169466  20.560  < 2e-16 ***
## day          0.1679981  0.0082353  20.400  < 2e-16 ***
## FFMC         0.0778674  0.0097079   8.021 1.05e-15 ***
## DMC          0.0070202  0.0003699  18.978  < 2e-16 ***
## DC          -0.0027381  0.0001783 -15.353  < 2e-16 ***
## ISI         -0.0594518  0.0066379  -8.956  < 2e-16 ***
## temp         0.0390550  0.0042767   9.132  < 2e-16 ***
## RH          -0.0324781  0.0015424 -21.057  < 2e-16 ***
## wind         0.0297267  0.0103998   2.858  0.00426 **
## rain        -1.4696739  0.6578003  -2.234  0.02547 *
## Zero hurdle model coefficients (binomial with logit link):
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.4233588  3.2393489  -0.439   0.6604
## X            0.0333129  0.0550280   0.605   0.5449
## Y            0.0314153  0.1042976   0.301   0.7633
## month        0.0947083  0.1152949   0.821   0.4114
## day          0.0388131  0.0525928   0.738   0.4605
## FFMC        -0.0048194  0.0345801  -0.139   0.8892
## DMC          0.0003756  0.0026215   0.143   0.8861
## DC          -0.0005685  0.0013284  -0.428   0.6687
## ISI         -0.0095384  0.0289900  -0.329   0.7421
## temp         0.0253505  0.0306127   0.828   0.4076
## RH          -0.0011855  0.0094447  -0.126   0.9001
## wind         0.1331939  0.0655889   2.031   0.0423 *
## rain         0.9420355  1.9661785   0.479   0.6319
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 20
## Log-likelihood: -6758 on 26 Df
```

```
predict_hurdle <- predict(model_hurdle, newdata = test_hurdle, type = "zero")
#MAE
mae_hurdle <- mean(abs(predict_hurdle - test_hurdle$area))

#MAD
mad_hurdle <- median(abs(predict_hurdle - mean(test_hurdle$area)))

# Accurancy
actual_zero <- ifelse(test_hurdle$area == 0, 1, 0)
predicted <- ifelse(predict_hurdle > 0.5, 1, 0)
table(actual_zero, predicted)
```

```
##            predicted
## actual_zero  0  1
##           0 48 41
##           1 34 33
```

```
(62+9)/(80+62+9+5)
```

```
## [1] 0.4551282
```

```
mean(predicted!=test_hurdle$area)
```

```
## [1] 0.7628205
```

```r
# Compared
data.frame(
  cbind(mae_zim, mae_hurdle),
  cbind(mad_zim, mad_hurdle)
)
```

```
##   mae_zim mae_hurdle  mad_zim mad_hurdle
## 1 25.9363   15.92599 10.61392   10.58626
```

```r
mae_zim
```

```
## [1] 25.9363
```

```r
# Monte Carlo simulation
# We want to do variables selection, for count model, since we need the variables be general, we exclude the X, Y
, month, day because it can vary from place to place. We rank in RH, DMC, DC, temp, FFMC, ISI, wind and rain in o
rder.
# For zero-hurdle model, wind is the only model that help us.

# select the top three important variables
hurdle_model <- hurdle(area ~ FFMC + DMC + DC + ISI + temp + RH + wind + rain|wind, data = train_hurdle, family =
"negbin")
```

```
## Warning in optim(fn = countDist, gr = countGrad, par = c(start$count, if (dist
## == : 控制裡不明的名稱 family
```

```
## Warning in optim(fn = zeroDist, gr = zeroGrad, par = c(start$zero, if
## (zero.dist == : 控制裡不明的名稱 family
```

```r
summary(hurdle_model)
```

```
##
## Call:
## hurdle(formula = area ~ FFMC + DMC + DC + ISI + temp + RH + wind + rain |
##     wind, data = train_hurdle, family = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.1741 -0.9057 -0.8086 -0.3638 36.8208
##
## Count model coefficients (truncated poisson with log link):
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.523e+00  6.875e-01  -2.216  0.02669 *
## FFMC         5.765e-02  7.734e-03   7.454 9.03e-14 ***
## DMC          5.461e-03  3.231e-04  16.902  < 2e-16 ***
## DC           1.282e-04  9.519e-05   1.346  0.17822
## ISI         -7.402e-02  5.911e-03 -12.521  < 2e-16 ***
## temp         2.646e-02  3.871e-03   6.836 8.14e-12 ***
## RH          -3.574e-02  1.470e-03 -24.314  < 2e-16 ***
## wind         1.055e-01  9.317e-03  11.326  < 2e-16 ***
## rain        -1.830e+00  6.573e-01  -2.784  0.00537 **
## Zero hurdle model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.5799     0.2586  -2.242   0.0249 *
## wind          0.1203     0.0598   2.012   0.0442 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 17
## Log-likelihood: -7347 on 11 Df
```

```r
# Remove some variables p-value larger than 2^e-10 <- DC and rain

# simulation preprocess
set.seed(4052)
sim_FFMC <- rnorm(n = 1000, mean = mean(hurdle_fires$FFMC), sd = sd(hurdle_fires$FFMC))
sim_DMC <- rnorm(n = 1000, mean = mean(hurdle_fires$DMC), sd = sd(hurdle_fires$DMC))
sim_ISI <- rnorm(n = 1000, mean = mean(hurdle_fires$ISI), sd = sd(hurdle_fires$ISI))
sim_temp <- rnorm(n = 1000, mean = mean(hurdle_fires$temp), sd = sd(hurdle_fires$temp))
sim_RH <- rnorm(n = 1000, mean = mean(hurdle_fires$RH), sd = sd(hurdle_fires$RH))
sim_wind <- rnorm(n = 1000, mean = mean(hurdle_fires$wind), sd = sd(hurdle_fires$wind))


while(any(sim_FFMC < 18)){
  sim_FFMC[sim_FFMC < 18] <- rnorm(sum(sim_FFMC < 18), mean = mean(hurdle_fires$FFMC), sd = sd(hurdle_fires$FFMC)
)
}
while(any(sim_DMC < 1)){
  sim_DMC[sim_DMC < 1] <- rnorm(sum(sim_DMC < 1), mean = mean(hurdle_fires$DMC), sd = sd(hurdle_fires$DMC))
}
while(any(sim_ISI < 0)){
  sim_ISI[sim_ISI < 0] <- rnorm(sum(sim_ISI < 0), mean = mean(hurdle_fires$ISI), sd = sd(hurdle_fires$ISI))
}
while(any(sim_temp < 2)){
  sim_temp[sim_temp < 2] <- rnorm(sum(sim_temp < 2), mean = mean(hurdle_fires$temp), sd = sd(hurdle_fires$temp))
}
while(any(sim_RH < 15)){
  sim_RH[sim_RH < 15] <- rnorm(sum(sim_RH < 15), mean = mean(hurdle_fires$RH), sd = sd(hurdle_fires$RH))
}
while(any(sim_wind < 0)){
  sim_wind[sim_wind < 0] <- rnorm(sum(sim_wind < 0), mean = mean(hurdle_fires$wind), sd = sd(hurdle_fires$wind))
}

sim_wind <- round(sim_wind, digits = 1)
sim_RH <- round(sim_RH, digits = 0)
sim_FFMC <- round(sim_FFMC, digits = 1)

sim_data <- data.frame(FFMC = sim_FFMC, DMC = sim_DMC, ISI = sim_ISI, temp = sim_temp, RH = sim_RH, wind = sim_wi
nd)


# Simulation
# training and testing set
model_hurdle <- hurdle(area ~ FFMC + DMC + ISI + temp + RH + wind|wind, data = train_hurdle, family = "negbin")
```

```
## Warning in optim(fn = countDist, gr = countGrad, par = c(start$count, if (dist
## == : 控制裡不明的名稱 family

## Warning in optim(fn = countDist, gr = countGrad, par = c(start$count, if (dist
## == : 控制裡不明的名稱 family
```

```r
summary(model_hurdle)
```

```
##
## Call:
## hurdle(formula = area ~ FFMC + DMC + ISI + temp + RH + wind | wind, data = train_hurdle,
##     family = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.1675 -0.9056 -0.8090 -0.3652 37.2787
##
## Count model coefficients (truncated poisson with log link):
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.2064714  0.6847403  -1.762   0.0781 .
## FFMC         0.0556419  0.0077364   7.192 6.38e-13 ***
## DMC          0.0057589  0.0002734  21.065  < 2e-16 ***
## ISI         -0.0714257  0.0058772 -12.153  < 2e-16 ***
## temp         0.0246154  0.0038183   6.447 1.14e-10 ***
## RH          -0.0369996  0.0014414 -25.668  < 2e-16 ***
## wind         0.0967280  0.0089627  10.792  < 2e-16 ***
## Zero hurdle model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.5799     0.2586  -2.242   0.0249 *
## wind          0.1203     0.0598   2.012   0.0442 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 15
## Log-likelihood: -7361 on 9 Df
```

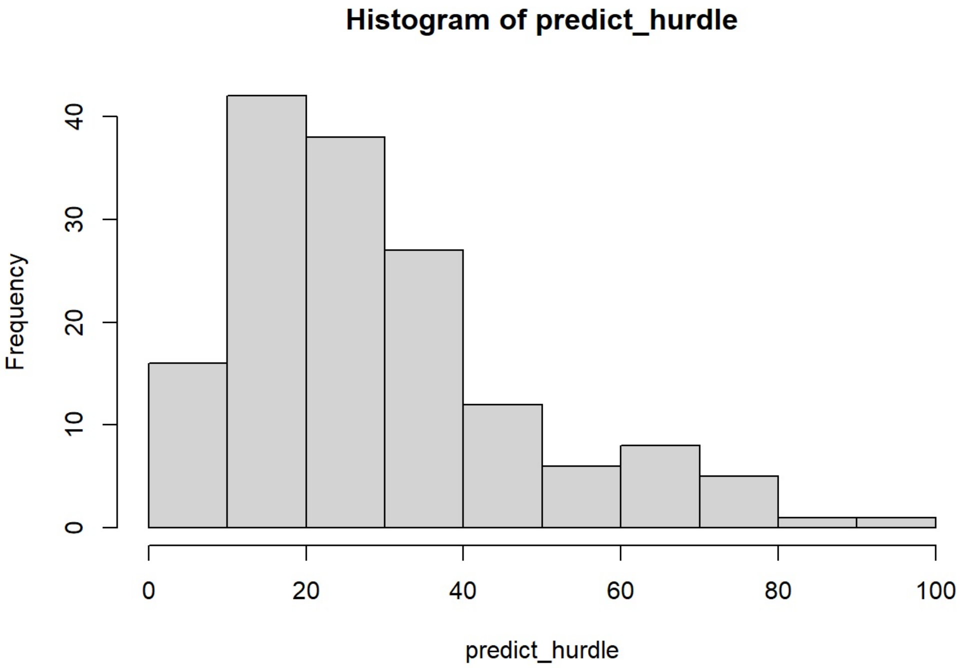
```
predict_hurdle <- predict(model_hurdle, newdata = test_hurdle, type = "count")

# simulation set
hurdle_model <- hurdle(area ~ FFMC + DMC + ISI + temp + RH + wind|wind, data = train_hurdle, family = "negbin")
```

```
## Warning in optim(fn = countDist, gr = countGrad, par = c(start$count, if (dist
## == : 控制裡不明的名稱 family

## Warning in optim(fn = countDist, gr = countGrad, par = c(start$count, if (dist
## == : 控制裡不明的名稱 family
```

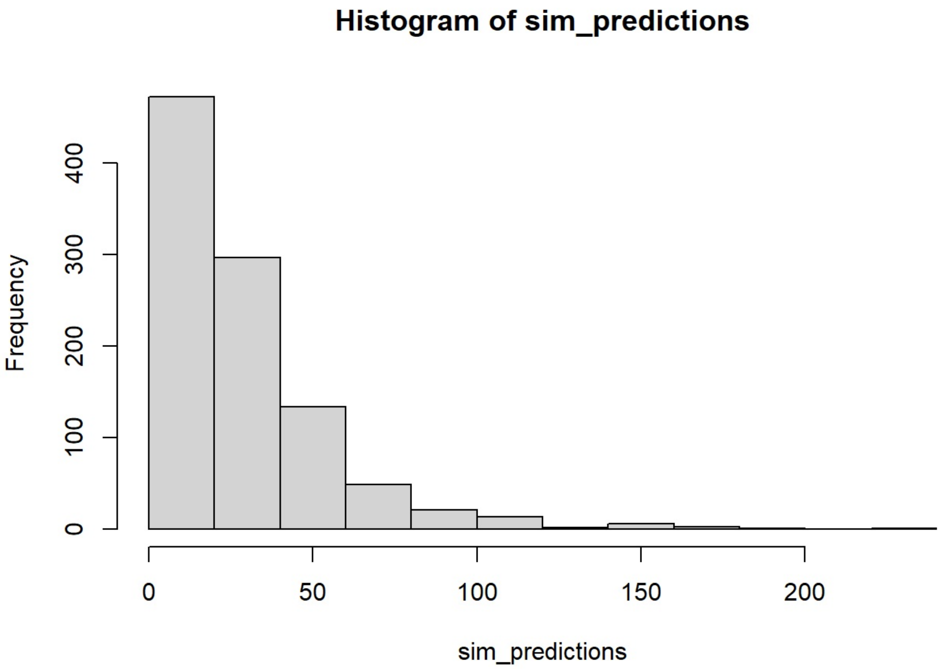
```
summary(hurdle_model)
```

```
##
## Call:
## hurdle(formula = area ~ FFMC + DMC + ISI + temp + RH + wind | wind, data = train_hurdle,
##     family = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.1675 -0.9056 -0.8090 -0.3652 37.2787
##
## Count model coefficients (truncated poisson with log link):
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.2064714  0.6847403  -1.762   0.0781 .
## FFMC         0.0556419  0.0077364   7.192 6.38e-13 ***
## DMC          0.0057589  0.0002734  21.065  < 2e-16 ***
## ISI         -0.0714257  0.0058772 -12.153  < 2e-16 ***
## temp         0.0246154  0.0038183   6.447 1.14e-10 ***
## RH          -0.0369996  0.0014414 -25.668  < 2e-16 ***
## wind         0.0967280  0.0089627  10.792  < 2e-16 ***
## Zero hurdle model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.5799     0.2586  -2.242   0.0249 *
## wind          0.1203     0.0598   2.012   0.0442 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 15
## Log-likelihood: -7361 on 9 Df
```

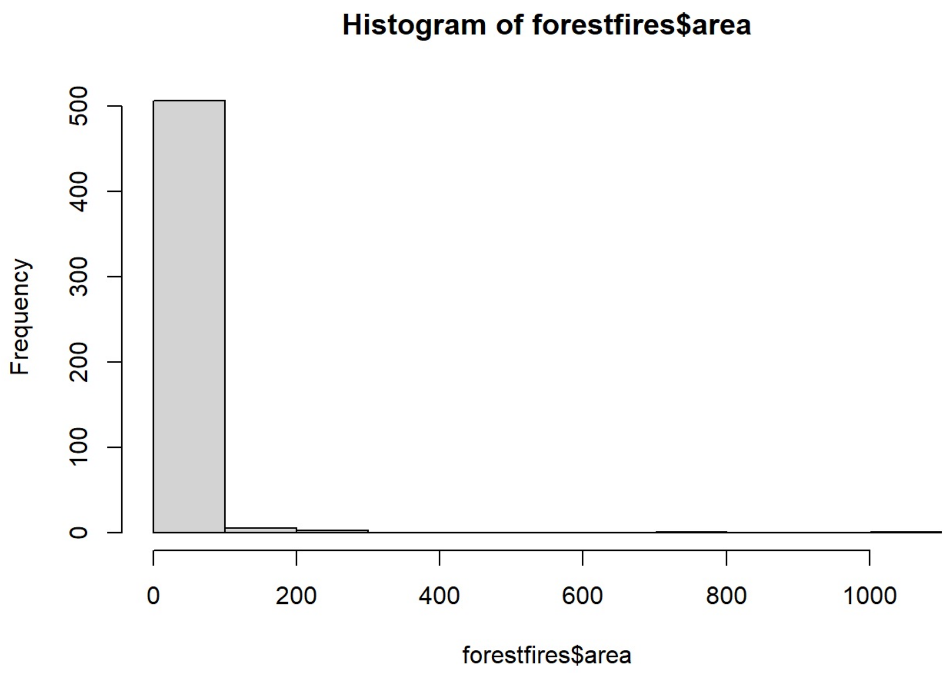
```
sim_predictions <- predict(hurdle_model, newdata = sim_data, type = "count")

# Compare.
hist(predict_hurdle)
```




```
hist(sim_predictions)
```




```
hist(forestfires$area)
```




```
# I like the result because the plot looks like the real world situation.
```

```
# 4052 single decision tree
library(MASS)
```

```
## Warning: 套件 'MASS' 是用 R 版本 4.3.3 來建造的
```

```
##
## 載入套件：'MASS'
```

```
## 下列物件被遮斷自 'package:glm2':
##
##     crabs
```

```
## 下列物件被遮斷自 'package:dplyr':
##
##     select
```

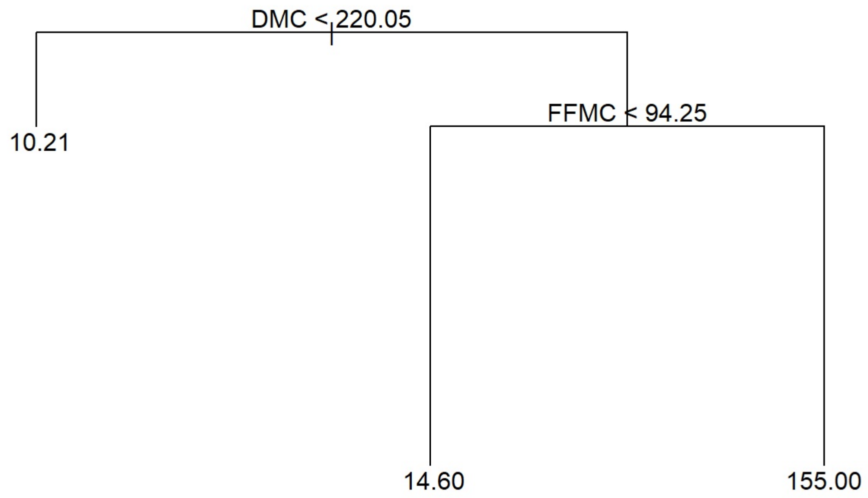
```
library(tree)
```

```
## Warning: 套件 'tree' 是用 R 版本 4.3.2 來建造的
```

```
dat = forestfires

nr = nrow(dat)
train_idr = sample(nr, nr*0.7)
trainr = dat[train_idr,]
valr = dat[-train_idr,]

m1r = tree(area ~ ., trainr)
summary(m1r)
```

```
##
## Regression tree:
## tree(formula = area ~ ., data = trainr)
## Variables actually used in tree construction:
## [1] "DMC"  "FFMC"
## Number of terminal nodes:  3
## Residual mean deviance:  2140 = 766000 / 358
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -155.00  -10.21   -9.67    0.00   -3.77  591.30
```

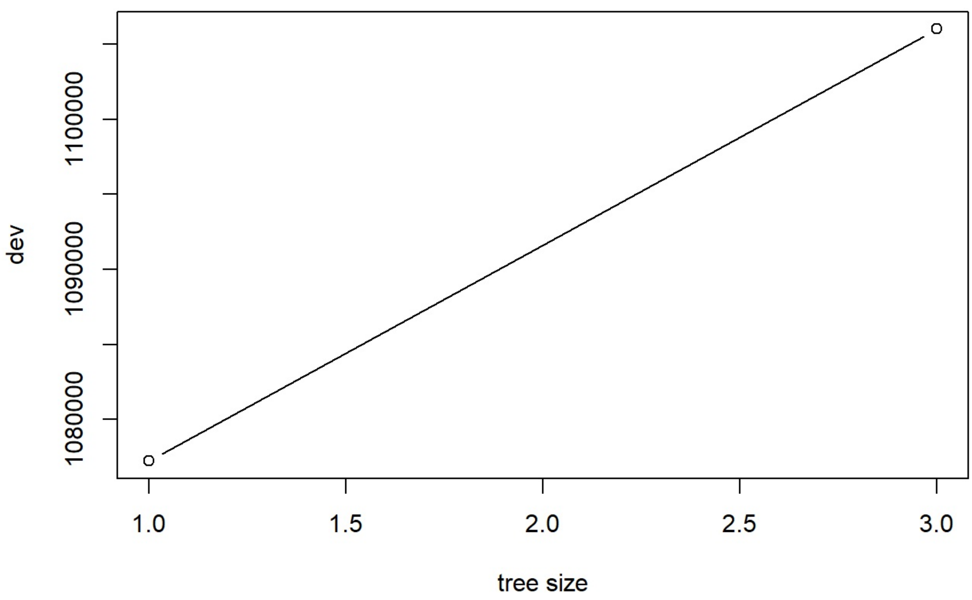
```
plot(m1r)
text(m1r, pretty = 0)
```



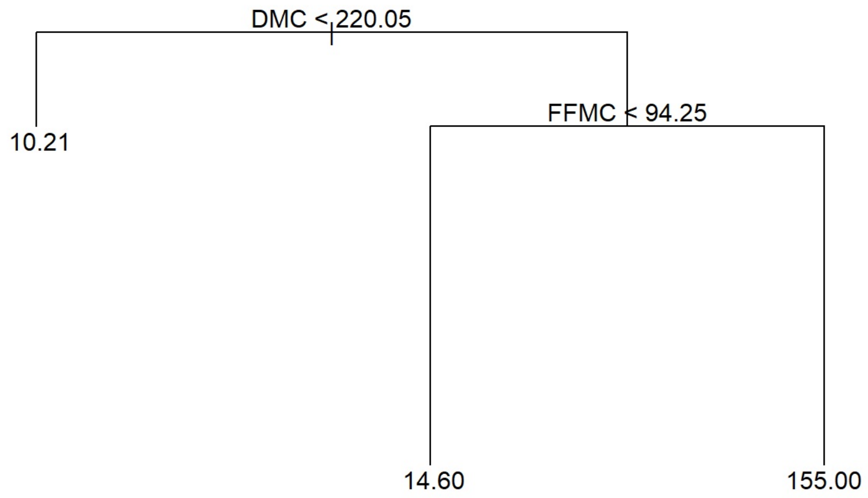

```
pred1r = predict(m1r, valr)
# Validation MSE
mean((pred1r - valr$area) ^ 2)
```

```
## [1] 8151.47
```

```
# find the optimal size to prune the tree
# Regression tree model
set.seed(4052)
# Cross-validation
m2r = cv.tree(m1r)

plot(m2r$size, m2r$dev, type = "b", xlab = "tree size", ylab = "dev")
```




```
m3r = prune.tree(m1r, best = 3)
plot(m3r)
text(m3r, pretty = 0)
```




```
pred3r = predict(m3r, valr)
mean((pred3r - valr$area) ^ 2)
```

```
## [1] 8151.47
```

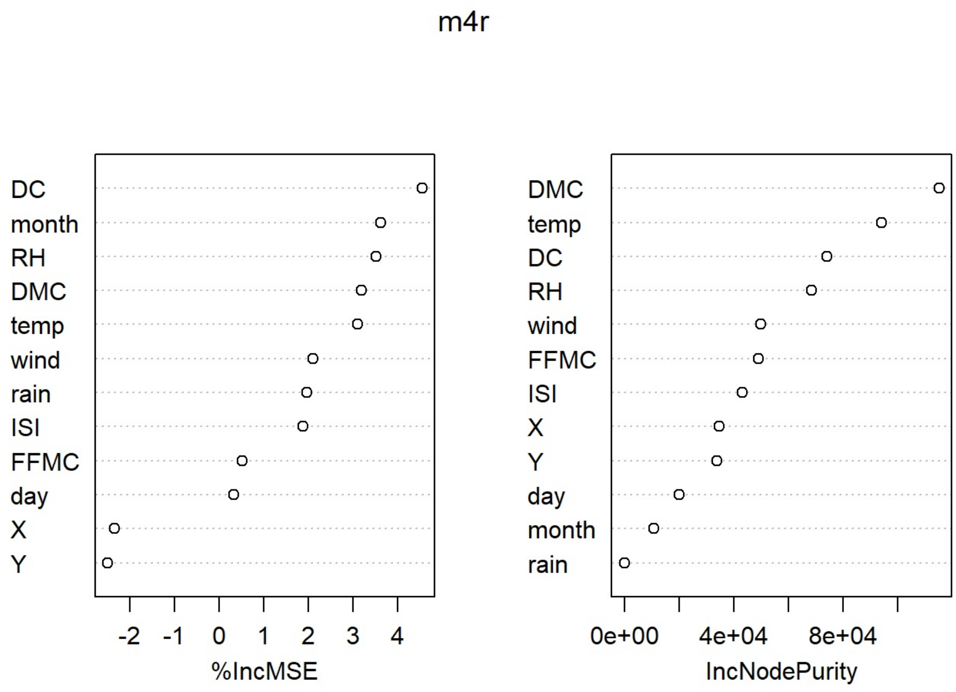
```
# RF and Bagging
# random forest model
m4r = randomForest(area ~ ., data = trainr, mtry = 3, importance = TRUE)

pred4r = predict(m4r, valr)
mean((pred4r - valr$area) ^ 2)
```

```
## [1] 7668.161
```

```
importance(m4r)
```

```
##          %IncMSE IncNodePurity
## X     -2.3469407    34619.3419
## Y     -2.5048927    33769.3940
## month  3.6102019    10746.6626
## day    0.3296131    19921.6444
## FFMC   0.5184158    49147.2549
## DMC    3.1889581   115106.0078
## DC     4.5416335    74134.9241
## ISI    1.8748322    43206.1472
## temp   3.1066654    94055.4296
## RH     3.5057927    68441.3878
## wind   2.0993694    49765.6147
## rain   1.9696371       51.5923
```

```
varImpPlot(m4r)
```




```
m4r
```

```
##
## Call:
##  randomForest(formula = area ~ ., data = trainr, mtry = 3, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 2687.796
##                     % Var explained: -11.61
```

```
# Gradient Boosting
library(gbm)
```

```
## Warning: 套件 'gbm' 是用 R 版本 4.3.3 來建造的
```

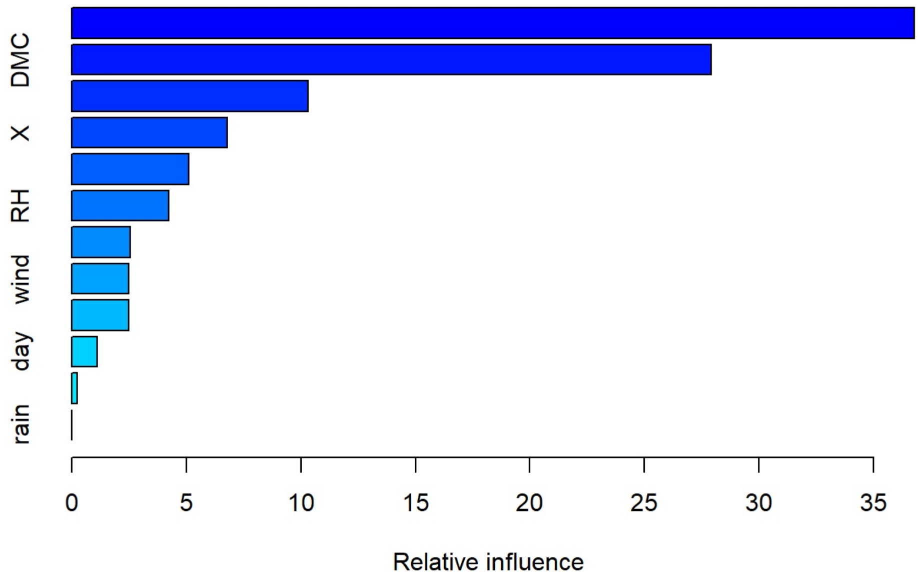
```
## Loaded gbm 2.1.9
```

```
## This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.com/gbm-dev
elopers/gbm3
```

```
m6r = gbm(area ~ ., data = trainr, distribution = "gaussian", n.trees = 5000, interaction.depth = 1, shrinkage =
0.1)

pred6r = predict(m6r, valr, n.trees = 5000)
mean((pred6r - valr$area) ^ 2)
```

```
## [1] 8052.724
```

```
summary(m6r)
```




```
##         var    rel.inf
## temp   temp 36.7685859
## DMC     DMC 27.8942679
## FFMC   FFMC 10.3115527
## X         X  6.8003648
## Y         Y  5.1176494
## RH       RH  4.2420036
## ISI     ISI  2.5427806
## wind   wind  2.4878756
## DC       DC  2.4874422
## day     day  1.1152938
## month month  0.2321836
## rain   rain  0.0000000
```

```
# MAE
mae_rf_lecture <- mean(abs(pred4r - valr))
```

```
## Warning in mean.default(abs(pred4r - valr)): 引數不是數值也不是邏輯值：回覆 NA
```

```
mae_rf_lecture
```

```
## [1] NA
```

```
# MAD for the test value
mad_rf_lecture <- mean(abs(pred4r - mean(pred4r)))
mad_rf_lecture
```

```
## [1] 6.081923
```

```r
# Random Forest
rf_fires <- forestfires

# Separate the data
set.seed(4893)

rf_fires = rf_fires[5:13]
nr = nrow(rf_fires)
train_indices = sample(nr, nr*0.7)
train_rf <- rf_fires[train_indices, ]
test_rf <- rf_fires[-train_indices, ]


# Use 10-fold cross-validation to find the optimal trees for the model
# Define the control using 10-fold cross-validation
train_control <- trainControl(method = "cv", number = 10)


# Train the model using the caret package to find the optimal mtry and ntree
# Note: the tuneGrid parameter can be used to specify custom values for mtry and ntree
model_rf_train <- train(area ~ ., data = train_rf, method = "rf",
               trControl = train_control,
               tuneLength = 20)
```

```
## note: only 7 unique complexity parameters in default grid. Truncating the grid to 7 .
```

```r
# tuneLength is the number of different values to try

# Print the results
model_rf_train$finalModel$mtry
```

```
## [1] 2
```

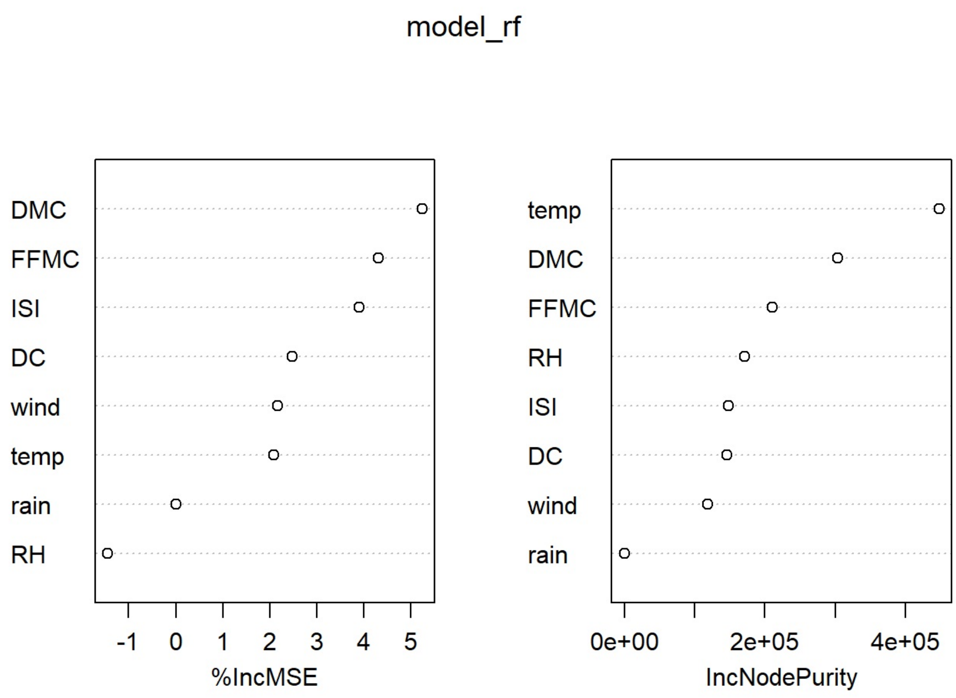
```r
model_rf_train$finalModel$ntree
```

```
## [1] 500
```

```r
# Building model
# The variance of explained is negative
# mtry = predictive variables / 3 = 13 / 3 = around 4
model_rf <- randomForest(area ~ ., data = train_rf, ntree = 500, mtry = 3, importance = TRUE)
prediction_rf <- predict(model_rf, test_rf)
importance(model_rf)
```

```
##         %IncMSE IncNodePurity
## FFMC  4.297895  210802.52114
## DMC   5.233123  303440.65081
## DC    2.471566  146132.71698
## ISI   3.888795  147666.09197
## temp  2.076321  447989.85184
## RH   -1.450419  171274.54750
## wind  2.158482  118221.59715
## rain  0.000000      69.93627
```

```r
varImpPlot(model_rf)
```




```
model_rf
```

```
##
## Call:
##  randomForest(formula = area ~ ., data = train_rf, ntree = 500,      mtry = 3, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 5689.648
##                     % Var explained: -5.65
```

```
# MAE
mae_rf <- mean(abs(prediction_rf - valr))
```

```
## Warning in mean.default(abs(prediction_rf - valr)):
## 引數不是數值也不是邏輯值：回覆 NA
```

```
mae_rf
```

```
## [1] NA
```

```
# MAD
mad_rf <- mean(abs(prediction_rf - mean(prediction_rf)))
mad_rf
```

```
## [1] 14.92574
```