pdp_boston.R

Kim Seok Joon

Thu Nov 01 22:29:55 2018

```
setwd("C:/Users/Kim Seok Joon/Desktop/연습")



library(randomForest)
```

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

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

```
library(pdp)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

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

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(RColorBrewer)
library(caret)
```

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

```
data("boston")
str(boston)
```

```
## 'data.frame':    506 obs. of  16 variables:
##  $ lon    : num  -71 -71 -70.9 -70.9 -70.9 ...
##  $ lat    : num  42.3 42.3 42.3 42.3 42.3 ...
##  $ cmedv  : num  24 21.6 34.7 33.4 36.2 28.7 22.9 22.1 16.5 18.9 ...
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : int  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ b      : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
```

```
data(boston)  # load the boston housing data
summary(boston)
```

```
##       lon              lat            cmedv            crim         
##  Min.   :-71.29   Min.   :42.03   Min.   : 5.00   Min.   : 0.00632  
##  1st Qu.:-71.09   1st Qu.:42.18   1st Qu.:17.02   1st Qu.: 0.08204  
##  Median :-71.05   Median :42.22   Median :21.20   Median : 0.25651  
##  Mean   :-71.06   Mean   :42.22   Mean   :22.53   Mean   : 3.61352  
##  3rd Qu.:-71.02   3rd Qu.:42.25   3rd Qu.:25.00   3rd Qu.: 3.67708  
##  Max.   :-70.81   Max.   :42.38   Max.   :50.00   Max.   :88.97620  
##        zn             indus       chas         nox               rm       
##  Min.   :  0.00   Min.   : 0.46   0:471   Min.   :0.3850   Min.   :3.561  
##  1st Qu.:  0.00   1st Qu.: 5.19   1: 35   1st Qu.:0.4490   1st Qu.:5.886  
##  Median :  0.00   Median : 9.69           Median :0.5380   Median :6.208  
##  Mean   : 11.36   Mean   :11.14           Mean   :0.5547   Mean   :6.285  
##  3rd Qu.: 12.50   3rd Qu.:18.10           3rd Qu.:0.6240   3rd Qu.:6.623  
##  Max.   :100.00   Max.   :27.74           Max.   :0.8710   Max.   :8.780  
##       age              dis              rad              tax       
##  Min.   :  2.90   Min.   : 1.130   Min.   : 1.000   Min.   :187.0  
##  1st Qu.: 45.02   1st Qu.: 2.100   1st Qu.: 4.000   1st Qu.:279.0  
##  Median : 77.50   Median : 3.207   Median : 5.000   Median :330.0  
##  Mean   : 68.57   Mean   : 3.795   Mean   : 9.549   Mean   :408.2  
##  3rd Qu.: 94.08   3rd Qu.: 5.188   3rd Qu.:24.000   3rd Qu.:666.0  
##  Max.   :100.00   Max.   :12.127   Max.   :24.000   Max.   :711.0  
##     ptratio            b              lstat      
##  Min.   :12.60   Min.   :  0.32   Min.   : 1.73  
##  1st Qu.:17.40   1st Qu.:375.38   1st Qu.: 6.95  
##  Median :19.05   Median :391.44   Median :11.36  
##  Mean   :18.46   Mean   :356.67   Mean   :12.65  
##  3rd Qu.:20.20   3rd Qu.:396.23   3rd Qu.:16.95  
##  Max.   :22.00   Max.   :396.90   Max.   :37.97  
```

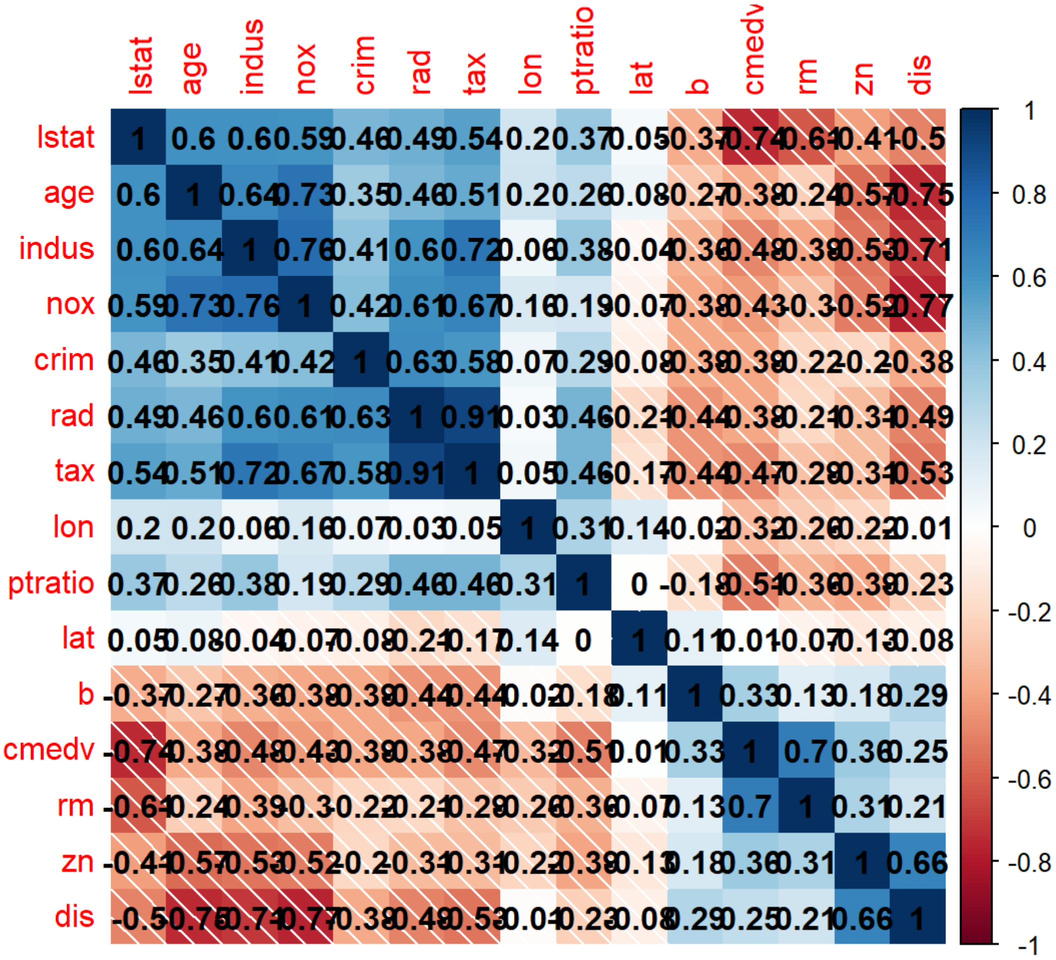
```
# x <- subset(boston, select=-chas)
# y <- boston$chas

##########################
###########EDA############
##########################
## 상관분석


head(boston)
```

```
##        lon     lat cmedv    crim zn indus chas   nox    rm  age    dis rad
## 1 -70.9550 42.2550  24.0 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1
## 2 -70.9500 42.2875  21.6 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2
## 3 -70.9360 42.2830  34.7 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2
## 4 -70.9280 42.2930  33.4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3
## 5 -70.9220 42.2980  36.2 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3
## 6 -70.9165 42.3040  28.7 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3
##   tax ptratio      b lstat
## 1 296    15.3 396.90  4.98
## 2 242    17.8 396.90  9.14
## 3 242    17.8 392.83  4.03
## 4 222    18.7 394.63  2.94
## 5 222    18.7 396.90  5.33
## 6 222    18.7 394.12  5.21
```

```
cor=subset(boston,select=-7)
head(cor)
```

```
##        lon     lat cmedv    crim zn indus   nox    rm  age    dis rad tax
## 1 -70.9550 42.2550  24.0 0.00632 18  2.31 0.538 6.575 65.2 4.0900   1 296
## 2 -70.9500 42.2875  21.6 0.02731  0  7.07 0.469 6.421 78.9 4.9671   2 242
## 3 -70.9360 42.2830  34.7 0.02729  0  7.07 0.469 7.185 61.1 4.9671   2 242
## 4 -70.9280 42.2930  33.4 0.03237  0  2.18 0.458 6.998 45.8 6.0622   3 222
## 5 -70.9220 42.2980  36.2 0.06905  0  2.18 0.458 7.147 54.2 6.0622   3 222
## 6 -70.9165 42.3040  28.7 0.02985  0  2.18 0.458 6.430 58.7 6.0622   3 222
##   ptratio      b lstat
## 1    15.3 396.90  4.98
## 2    17.8 396.90  9.14
## 3    17.8 392.83  4.03
## 4    18.7 394.63  2.94
## 5    18.7 396.90  5.33
## 6    18.7 394.12  5.21
```

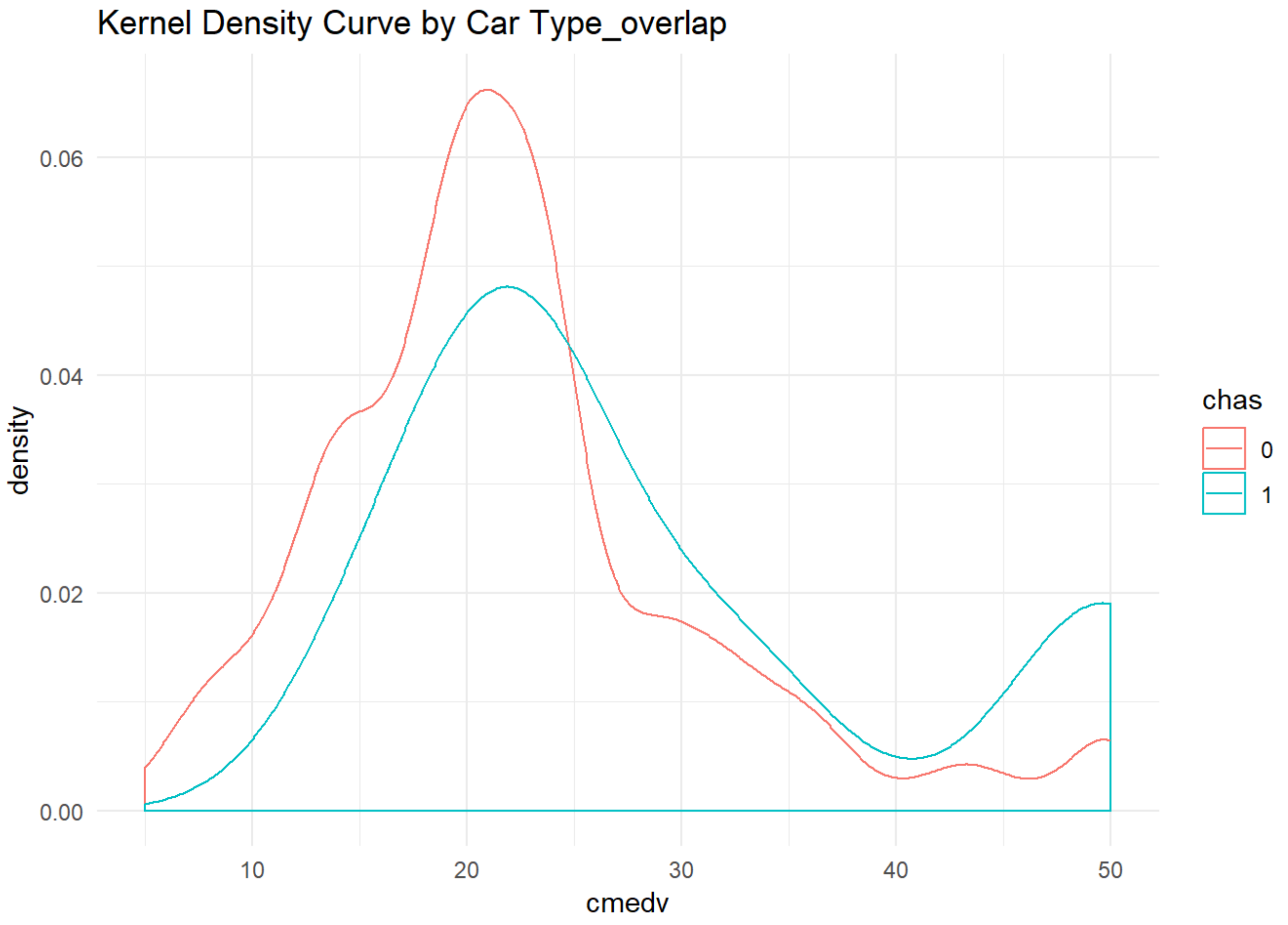
```
cor=cor[(complete.cases(cor)),]

corrplot <- cor(cor)
corrplot(corrplot, order = "hclust",addCoef.col="black", method="shade")
```



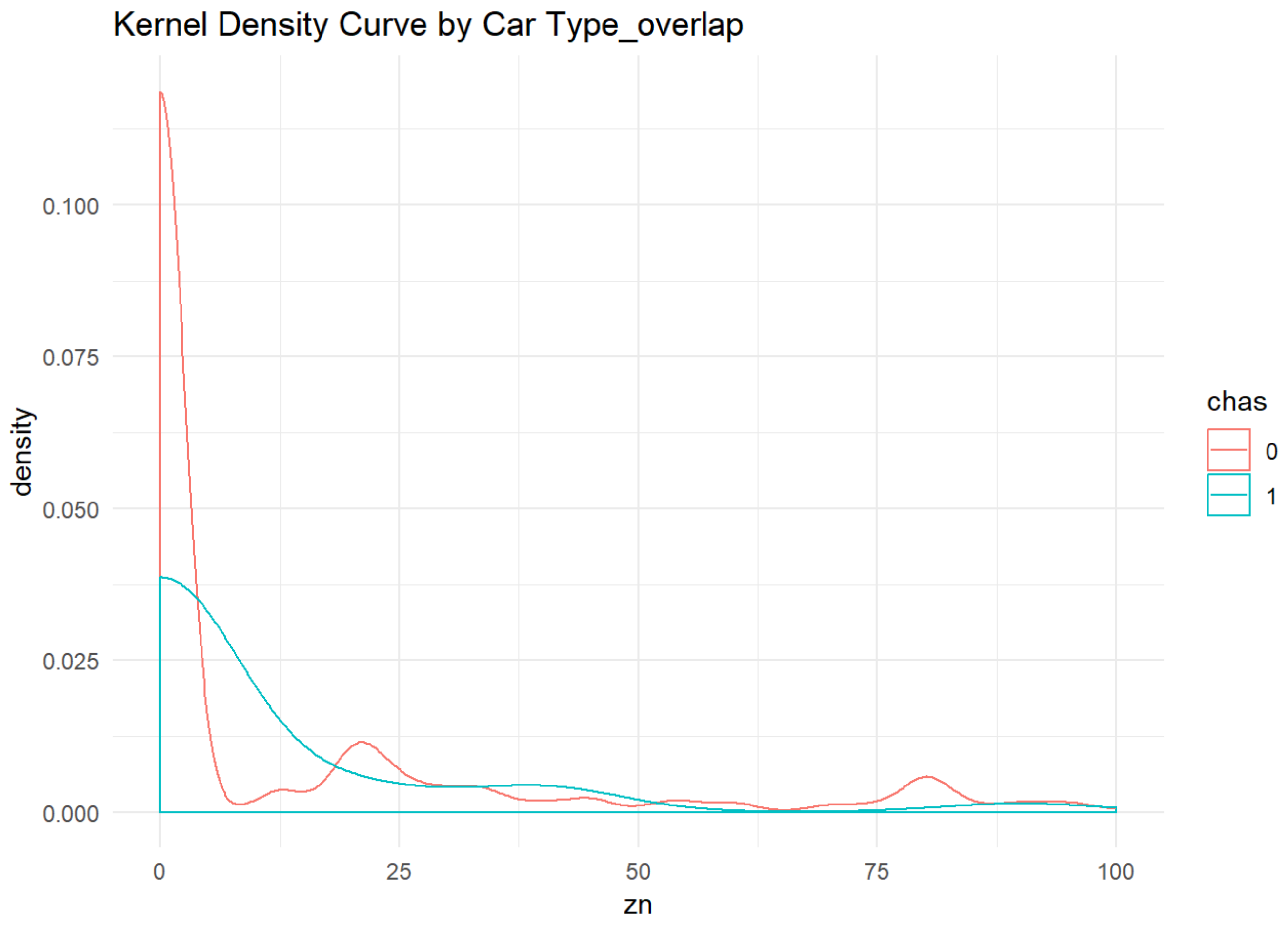
```
# 서로 속성이 거의 동일한 feature 다수 존재,
# 상관계수를 기준으로 유의한 feature 종류 중 하나만 남겨두고 나머진 제거
# feature간 상관계수 0.91 이상 제거

#################################
#Feature의 class별 밀도 그래프#
#################################
## -> 1. 클래스에 대한 설명력 파악
#          - 각 클래스의 특성파악으로 분류모델 학습 시 적절한 feature를 찾기 위함.
#          - 최빈값, 왜도, 첨도, 분산, 정규성
#      2. 모델에서의 feature engineering
#          - Feature의 분포(형태)가 분류모델에 적/합한지 판단하기 위함


# 왜도 VS 첨도
# 그럼 이 두개는 어떨 때 중요한게 쓰일까요?
#    바로 정규분포를 확인할 때 쓰입니다.

# 왜도는 좌우로 치우치는 정도를 말합니다.
# 왜도가 0이라면 좌우대칭하다고 하며
# >0이면 오른쪽으로 비대칭, <0이면 왼쪽으로 비대칭
# 하다고 말합니다.
#
# 첨도(최빈값의 빈도)는 위로 얼마나 뾰족한지 알 수 있는것인데
# 첨도는 양(+)일 경우 더 뾰족하고
# 음(-)일 경우 덜 뾰족한 편입니다.

head(boston)
```

```
##         lon     lat cmedv    crim zn indus chas   nox    rm  age    dis rad
## 1 -70.9550 42.2550  24.0 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1
## 2 -70.9500 42.2875  21.6 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2
## 3 -70.9360 42.2830  34.7 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2
## 4 -70.9280 42.2930  33.4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3
## 5 -70.9220 42.2980  36.2 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3
## 6 -70.9165 42.3040  28.7 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3
##   tax ptratio      b lstat
## 1 296    15.3 396.90  4.98
## 2 242    17.8 396.90  9.14
## 3 242    17.8 392.83  4.03
## 4 222    18.7 394.63  2.94
## 5 222    18.7 396.90  5.33
## 6 222    18.7 394.12  5.21
```

```
a=ggplot(boston, aes(x=cmedv, colour = chas)) +
  geom_density(fill = NA) +
  geom_line(stat = "density") +
  expand_limits(y = 0) +
  theme_minimal() +
  ggtitle("Kernel Density Curve by Car Type_overlap")
a
```



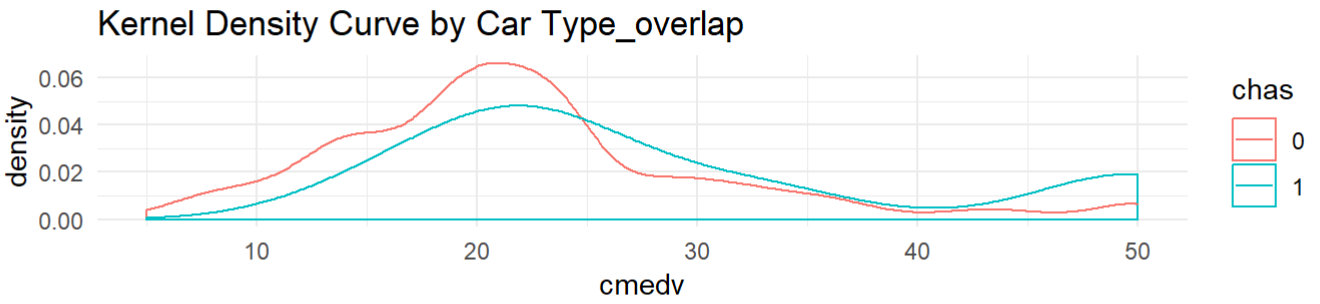
```
b=ggplot(boston, aes(x= zn, colour = chas)) +
  geom_density(fill = NA) +
  geom_line(stat = "density") +
  expand_limits(y = 0) +
  theme_minimal() +
  ggtitle("Kernel Density Curve by Car Type_overlap")
b
```

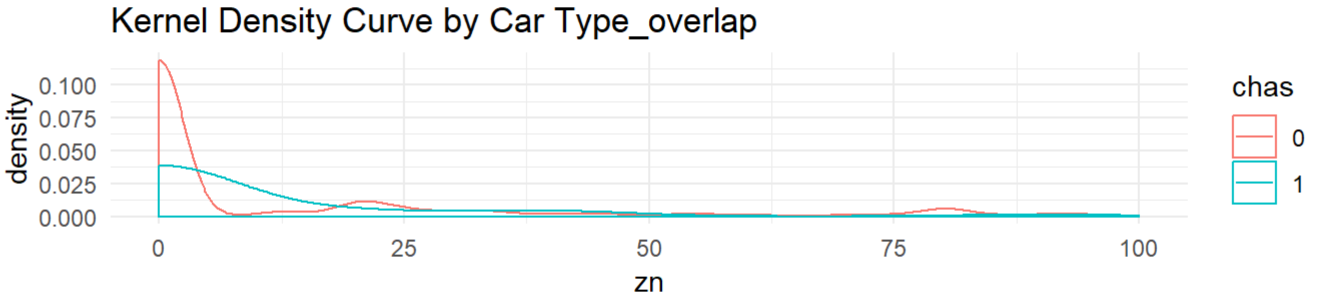


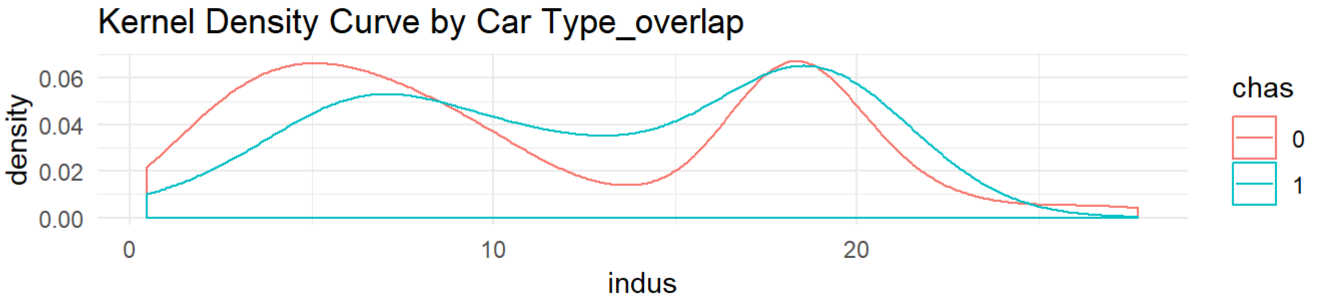
```
c=ggplot(boston, aes(x=indus, colour = chas)) +
  geom_density(fill = NA) +
  geom_line(stat = "density") +
  expand_limits(y = 0) +
  theme_minimal() +
  ggtitle("Kernel Density Curve by Car Type_overlap")
C
```

```
## function (object, contr, how.many, ...)
## {
##     if (!nlevels(object))
##         stop("object not interpretable as a factor")
##     if (!missing(contr) && is.name(Xcontr <- substitute(contr)))
##         contr <- switch(as.character(Xcontr), poly = "contr.poly",
##             helmert = "contr.helmert", sum = "contr.sum", treatment = "contr.treatment",
##             SAS = "contr.SAS", contr)
##     if (missing(contr)) {
##         oc <- getOption("contrasts")
##         contr <- if (length(oc) < 2L)
##             if (is.ordered(object))
##                 contr.poly
##             else contr.treatment
##         else oc[1 + is.ordered(object)]
##     }
##     if (missing(how.many) && missing(...))
##         contrasts(object) <- contr
##     else {
##         if (is.character(contr))
##             contr <- get(contr, mode = "function")
##         if (is.function(contr))
##             contr <- contr(nlevels(object), ...)
##         contrasts(object, how.many) <- contr
##     }
##     object
## }
## <bytecode: 0x000000001cf2e528>
## <environment: namespace:stats>
```

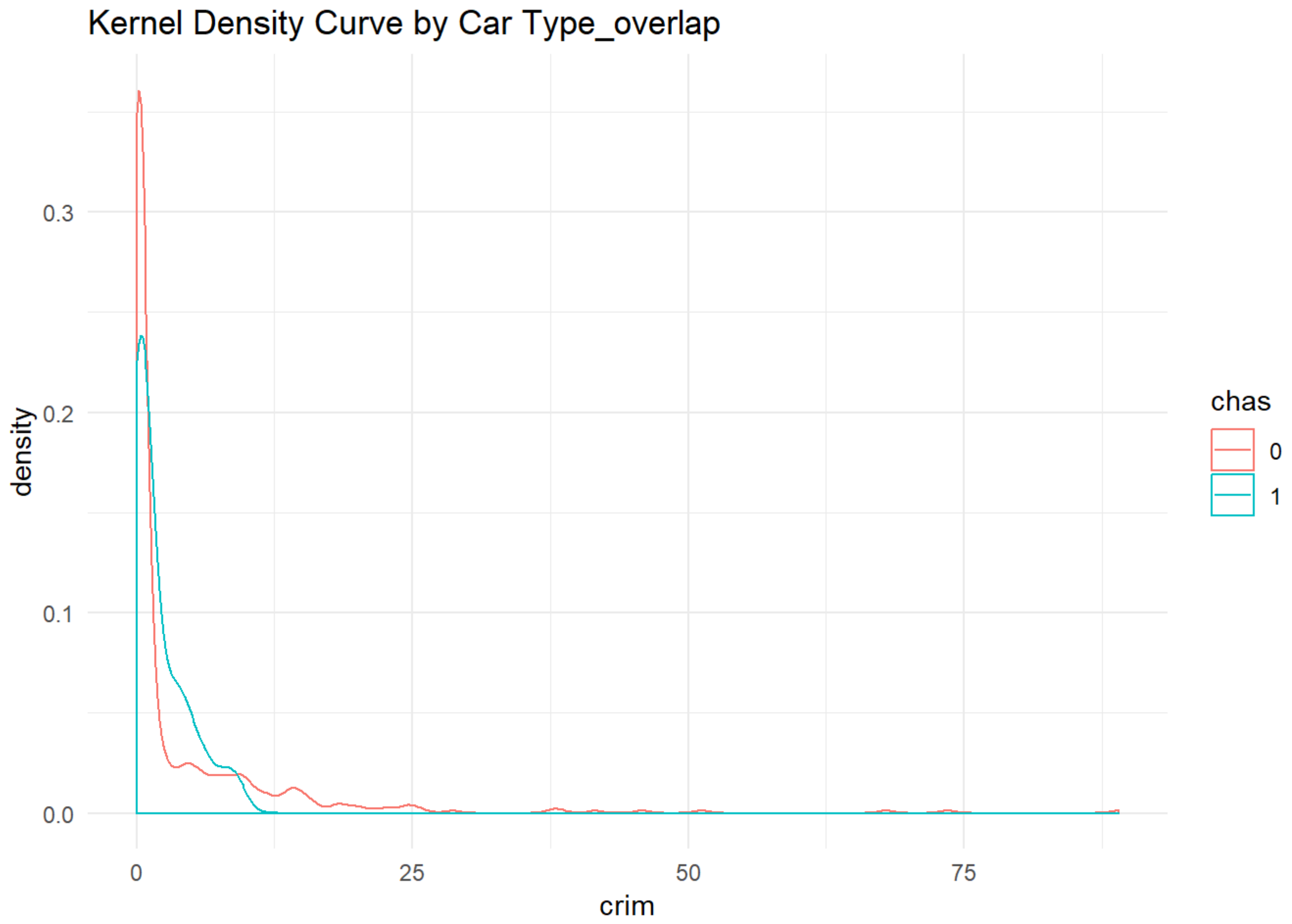
```
grid.arrange(a,b,c)
```







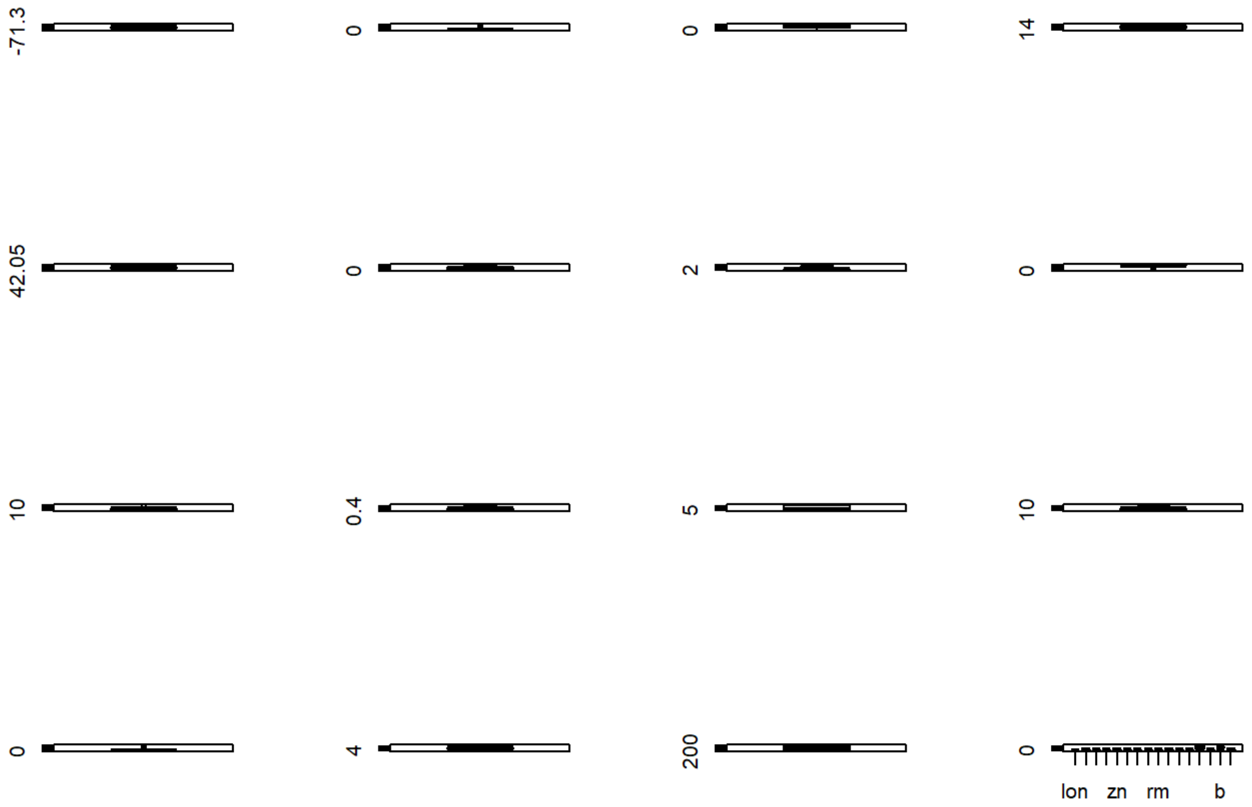
```
# 해당 feature들은 밀도 그래프에서 o,1 최빈값이 유사한 것을 볼 수 있지만,
# 최빈값의 빈도(첨도)에서 차이가 나는 것으로 보아 0,1 분류에 좋은 feature가 될 것으로 판단


d=ggplot(boston, aes(x=crim, colour = chas)) +
  geom_density(fill = NA) +
  geom_line(stat = "density") +
  expand_limits(y = 0) +
  theme_minimal() +
  ggtitle("Kernel Density Curve by Car Type_overlap")
d # 클래스간 밀도 그래프의 개형 차이가 극명한 경우
```


Kernel Density Curve by Car Type_overlap

```
# 밀도 그래프의 왜도
# 밀도 그래프의 첨도
# 밀도 그래프의 분산 등을 고려하였을 때 각 클래스를 분류할 때 좋은 feature가 될 것으로 판단


# 전처리
sum(is.na(boston))
```

```
## [1] 0
```

```
# 이상치
par(mfcol=c(4,4))

for (i in c(1:6,8:16)){
  boxplot(boston[i])}

boxplot(boston)
```



```
str(df)
```

```
## function (x, df1, df2, ncp, log = FALSE)
```

```
head(df)
```

```
##                                                  
## 1 function (x, df1, df2, ncp, log = FALSE)       
## 2 {                                              
## 3     if (missing(ncp))                          
## 4         .Call(C_df, x, df1, df2, log)          
## 5     else .Call(C_dnf, x, df1, df2, ncp, log)   
## 6 }
```

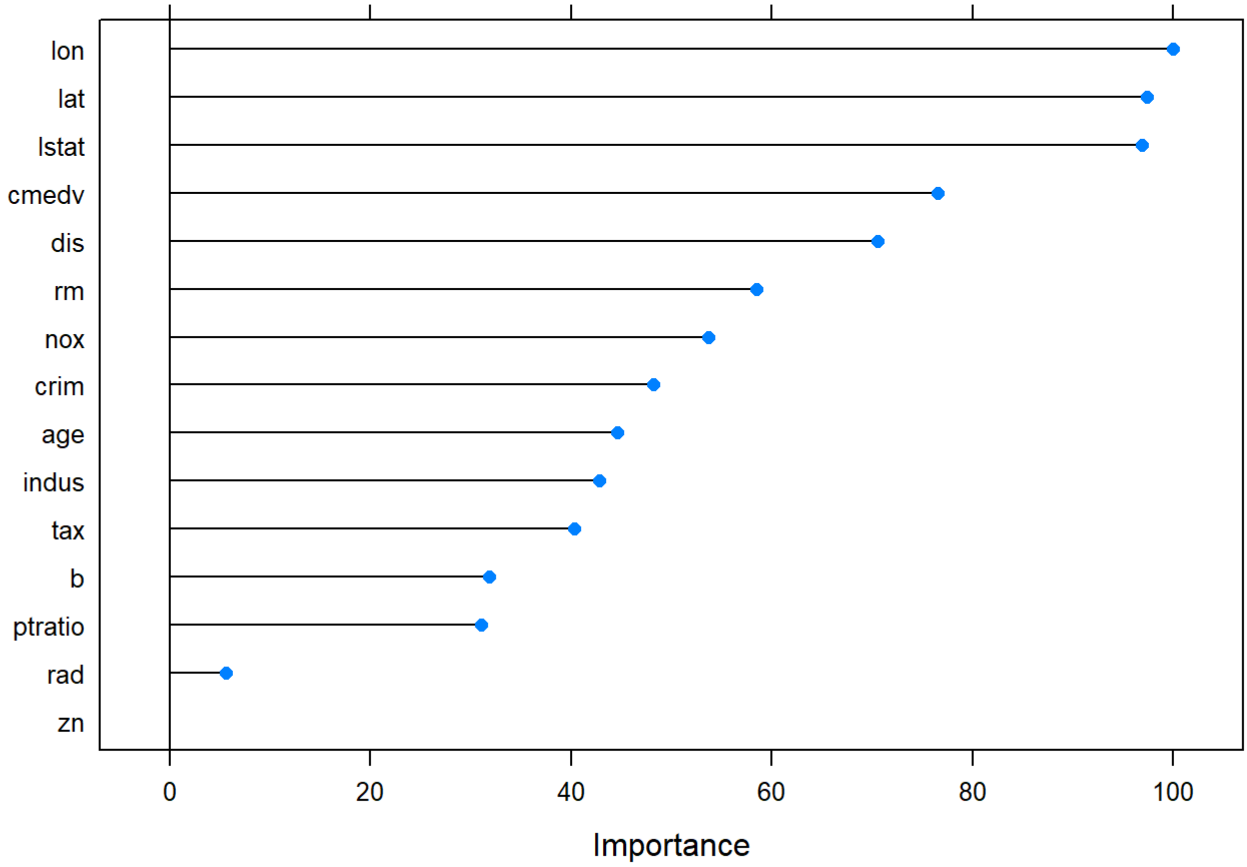
```
# 모델링

set.seed(1000)
intrain=createDataPartition(y=boston$chas, p =0.7, list=FALSE)
train = boston[intrain,]
test = boston[-intrain,]

rf=train(x=train[,-7], y=train$chas, data=train, method='rf', trControl=trainControl(method="cv", numb
er=5))

# Predicting
pred <- predict(rf, test, probability = TRUE)

# Confusion Matrix
confusionMatrix(pred,test$chas)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0 139   9
##          1   2   1
## 
##                Accuracy : 0.9272
##                  95% CI : (0.8734, 0.9631)
##     No Information Rate : 0.9338
##     P-Value [Acc > NIR] : 0.70078
## 
##                   Kappa : 0.1272
##  Mcnemar's Test P-Value : 0.07044
## 
##             Sensitivity : 0.9858
##             Specificity : 0.1000
##          Pos Pred Value : 0.9392
##          Neg Pred Value : 0.3333
##              Prevalence : 0.9338
##          Detection Rate : 0.9205
##    Detection Prevalence : 0.9801
##       Balanced Accuracy : 0.5429
## 
##        'Positive' Class : 0
## 
```

```
table(pred,test$chas)
```

```
## 
## pred   0   1
##    0 139   9
##    1   2   1
```

```
# 변수중요도
par(mfrow=c(1,1))
plot(varImp(rf))
```



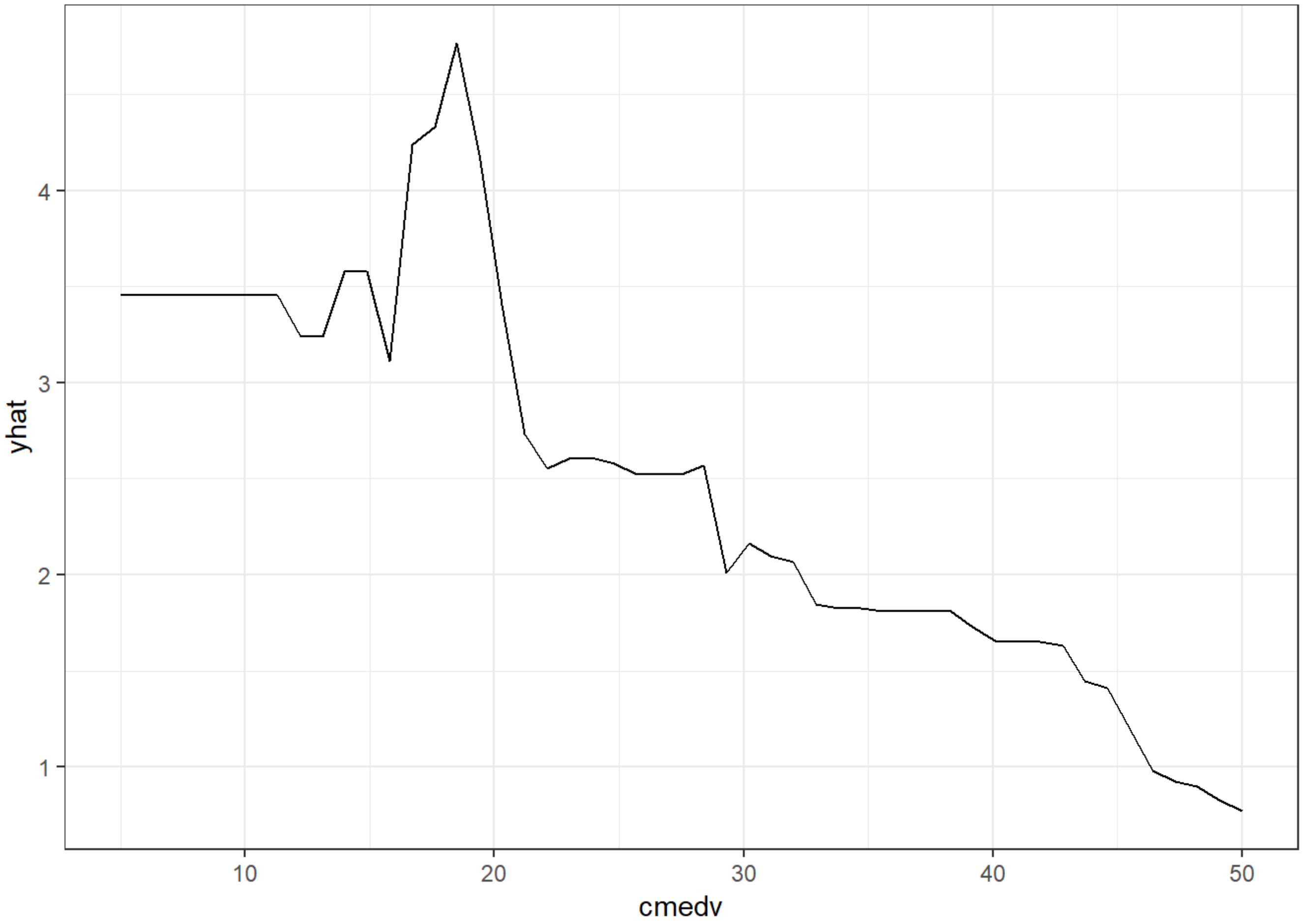
```
varimp=varImp(rf)
mean(varimp$importance$Overall)
```

```
## [1] 53.19743
```

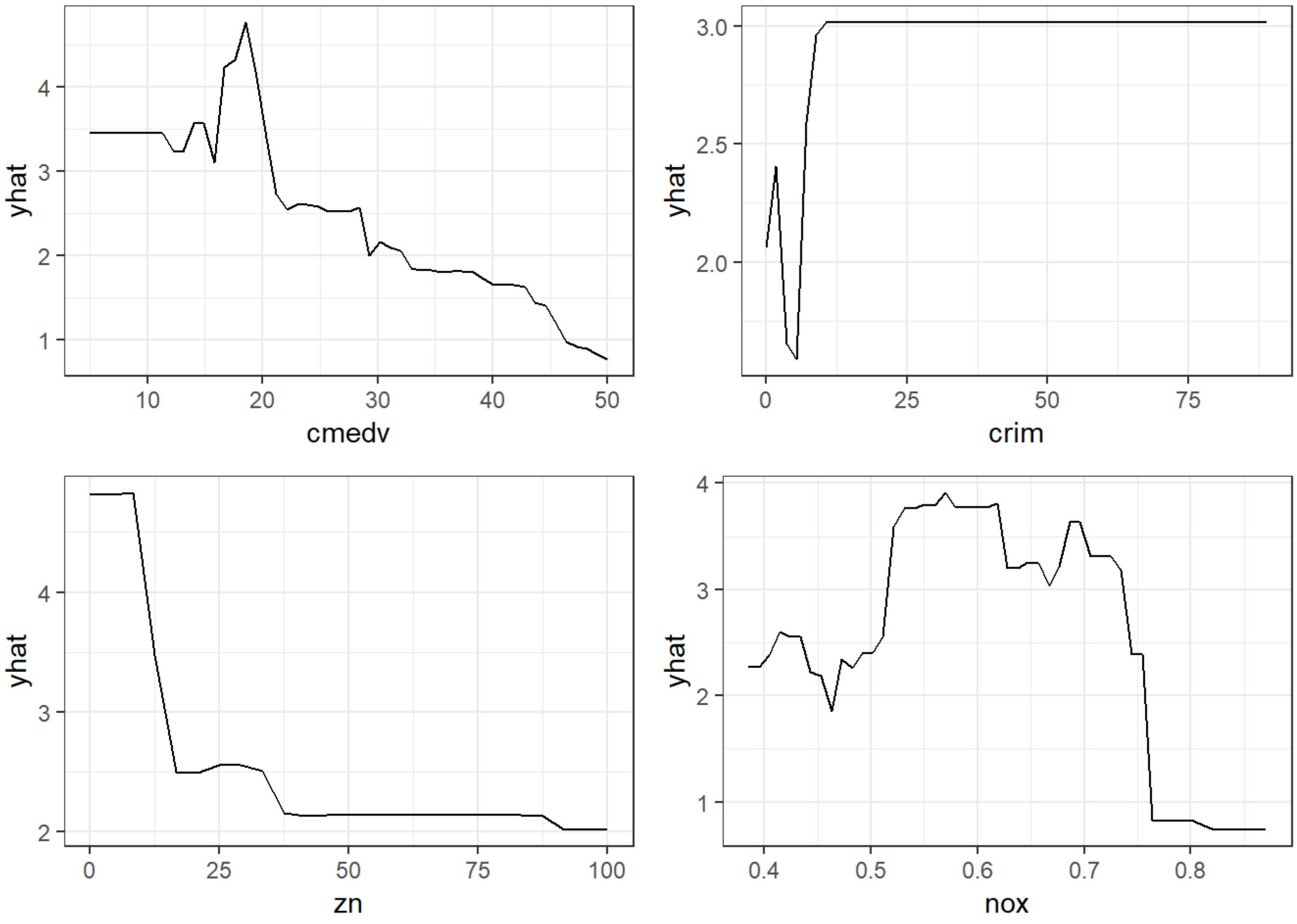
```
# which(varimp$importance >mean(varimp$importance$Overall)) # 중요도 평균이상만 출력

# rownames(as.data.frame(varimp$importance))[which(varimp$importance >mean(varimp$importance$Overal
l))]

##########################
###########PDP############
##########################
# 변수해석
# - 트리 모델에서 변수 중요성은 알 수 있지만 직접적인 상관성의 파악은 어려움.
# 이에 대한 대안적 탐색 => PD(Partial Dependence)
# 해당 feature의 값이 증가할수록 O(True Positive: 진양성) 분류 확률이 높아짐


# 단일 예측변수에 대한 PDP
# Single Variable
head(boston)
```

```
##        lon     lat cmedv    crim zn indus chas   nox    rm  age    dis rad
## 1 -70.9550 42.2550  24.0 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1
## 2 -70.9500 42.2875  21.6 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2
## 3 -70.9360 42.2830  34.7 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2
## 4 -70.9280 42.2930  33.4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3
## 5 -70.9220 42.2980  36.2 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3
## 6 -70.9165 42.3040  28.7 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3
##   tax ptratio      b lstat
## 1 296    15.3 396.90  4.98
## 2 242    17.8 396.90  9.14
## 3 242    17.8 392.83  4.03
## 4 222    18.7 394.63  2.94
## 5 222    18.7 396.90  5.33
## 6 222    18.7 394.12  5.21
```

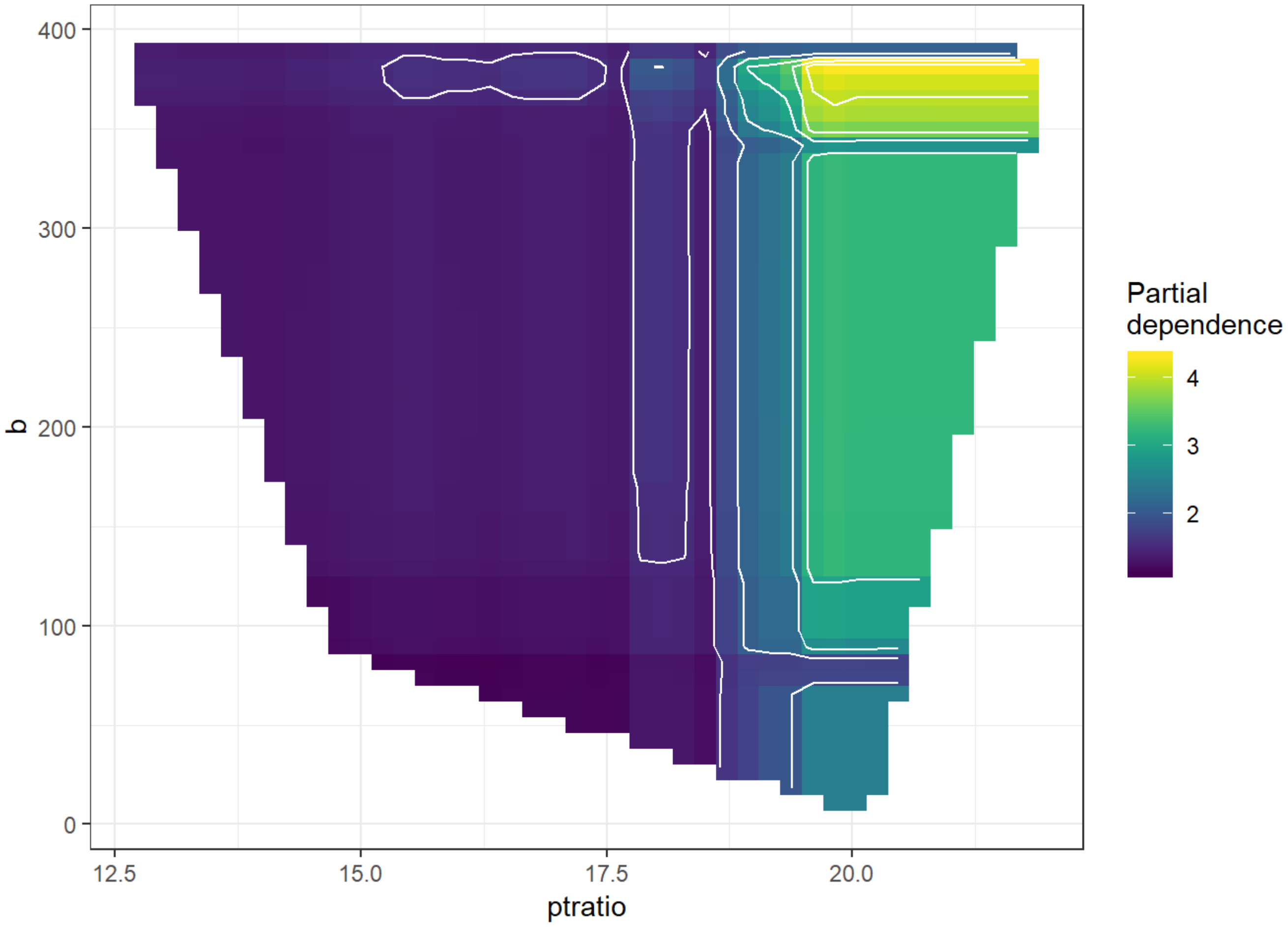
```
par.Petal_W <- partial(rf, pred.var = c("cmedv"), chull = TRUE)
plot.Petal_W <- autoplot(par.Petal_W, contour = TRUE)+theme_minimal()+theme_bw()
plot.Petal_W # cmedv 증가 => 0(True Positive: 진양성) 분류 확률 낮아짐
```



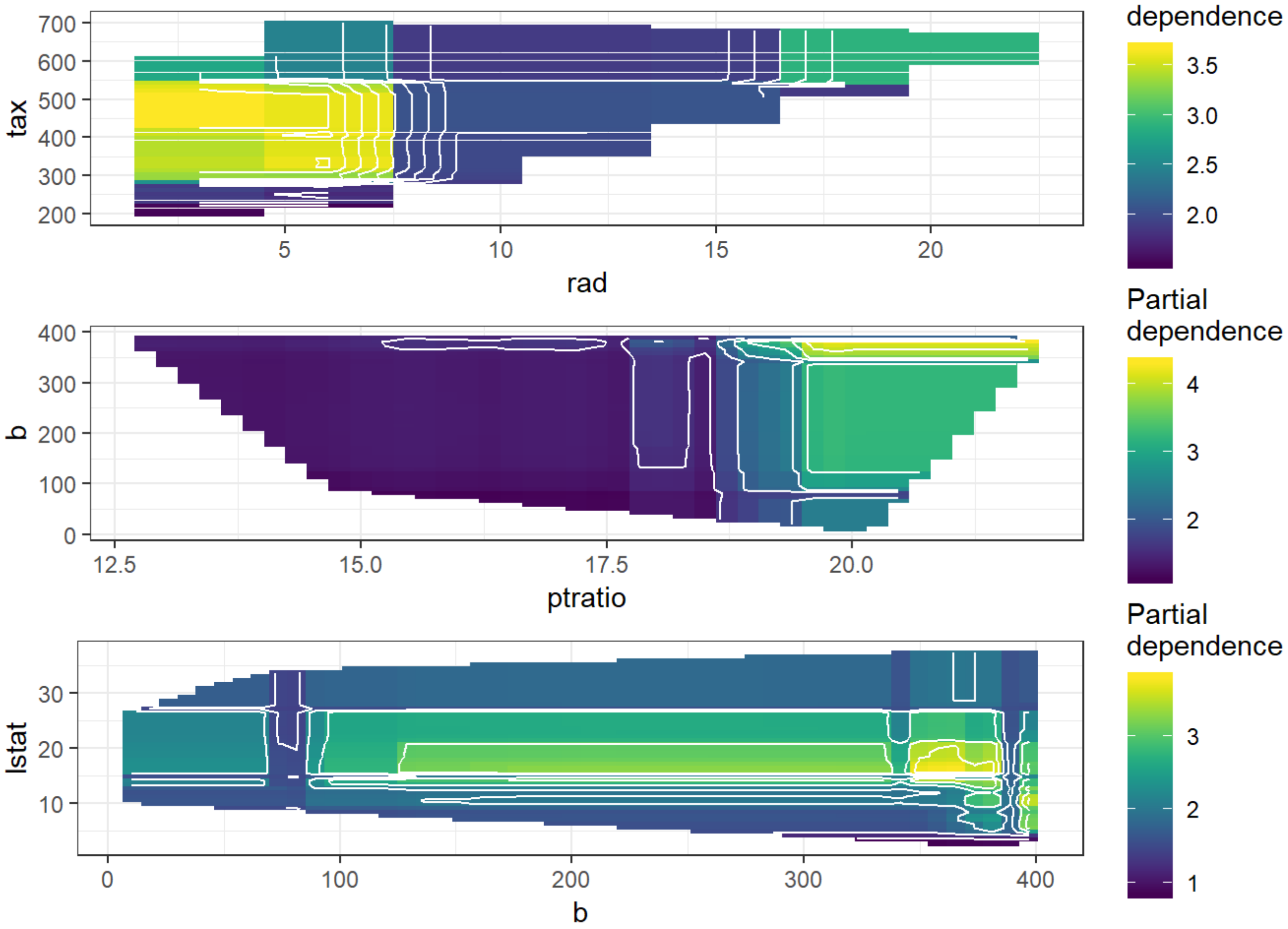
```
# Single Variable
par.Sepal_W  <- partial(rf, pred.var = c("crim"), chull = TRUE)
plot.Sepal_W  <- autoplot(par.Sepal_W , contour = TRUE)+theme_minimal()+theme_bw()

# Single Variable
par.Petal_L <- partial(rf, pred.var = c("zn"), chull = TRUE)
plot.Petal_L <- autoplot(par.Petal_L, contour = TRUE)+theme_minimal()+theme_bw()

# Single Variable
par.Sepal_L  <- partial(rf, pred.var = c("nox"), chull = TRUE)
plot.Sepal_L  <- autoplot(par.Sepal_L , contour = TRUE)+theme_minimal()+theme_bw()

grid.arrange(plot.Petal_W, plot.Sepal_W, plot.Petal_L, plot.Sepal_L)
```



```
# 2개 예측변수의 상호작용을 고려한 PDP
# Two Variables
par.Petal_W.Sepal_W1 <- partial(rf, pred.var = c("rad", "tax"), chull = TRUE)
plot.Petal_W.Sepal_W1 <- autoplot(par.Petal_W.Sepal_W1, contour = TRUE,
                                 legend.title = "Partial₩ndependence")

par.Petal_W.Sepal_W2 <- partial(rf, pred.var = c("ptratio", "b"), chull = TRUE)
plot.Petal_W.Sepal_W2 <- autoplot(par.Petal_W.Sepal_W2, contour = TRUE,
                                 legend.title = "Partial₩ndependence")
plot.Petal_W.Sepal_W2
```



```
# b, ptratio 증가 => 0(True Positive: 진양성) 분류 확률 높아짐

par.Petal_W.Sepal_W3 <- partial(rf, pred.var = c("b", "lstat"), chull = TRUE)
plot.Petal_W.Sepal_W3 <- autoplot(par.Petal_W.Sepal_W3, contour = TRUE,
                                  legend.title = "Partial₩ndependence")

grid.arrange(plot.Petal_W.Sepal_W1,plot.Petal_W.Sepal_W2,plot.Petal_W.Sepal_W3)
```


tax
700
600
500
400
300
200
5
10
15
20
rad
dependence
3.5
3.0
2.5
2.0
b
400
300
200
100
0
12.5
15.0
17.5
20.0
ptratio
Partial dependence
4
3
2
lstat
30
20
10
0
100
200
300
400
b
Partial dependence
3
2
1

```
##########################
###########도움말#########
##########################

# 결측치 처리 및 대치법
# # Missing value rate > 5% => Delete
# sum(is.na(df)) # 220779
# nrow(df[df$asile_type=="",])/28663 *100 # 63.39532%
# nrow(df[df$earthquake=="",])/28663 *100 # 23.71001%
#
# pMiss <- function(x){sum(is.na(x))/length(x)*100}
# apply(df, 2, pMiss)
# df=subset(df,select = -c(asile_type,building_count, building_coverage_ratio,commute_dmc, commute_seo
ngsu
#                          ,commute_yongsan, commute_chungmuro, earthquake, floor_area_ratio, floor_ma
x, floor_min
#                          ,parking_inside,parking_outside,parking_rate,permission_date, slope))
#
# apply(df, 1, pMiss) < 5
# df=df[apply(df, 1, pMiss) < 5,]
#
# # Dummy variable
# df=dummy.data.frame(df)
#
# # Multiple imputation
# imp = mice(df, m=5, maxit = 50, seed=1234, method='cart')
# df=complete(imp)


######################################################################################
# ####전처리####
#
# library(dummies)
# library(xgboost)
# library(mice)
# library(caret)
# library(dplyr)
# library(ggplot2)
# library(corrplot)
# library(RColorBrewer)
# library(caret)
# library(car)
# library(ROSE)
# library(rpart)
# library(MASS)
# library(e1071)
# library(glmnet)
#
# # 예측변수 생성
# df$accident = ifelse(is.na(df$관할해경서),0,1) # 발생X:'0',발생O:'1'
# df$accident=as.factor(df$accident)
#
#
# # 변수 분리 및 제거
# df$year=as.factor(substr(df$일시,1,4))
# df$month=as.factor(substr(df$일시,6,7))
# df$day=as.factor(substr(df$일시,9,10))
# df$hour=substr(df$일시,12,13)
# df$hour=as.factor(gsub(":","",df$hour))
# df$일시=as.POSIXct(df$일시, format="%Y-%m-%d %H:%M")
# df[1]=NULL
# df[15:22]=NULL
# df$week=strftime(df$일시, '%u') # 요일
#
# #######################################################################################
# ####시계열그래프####
#
# # 연도별 사고 데이터 시계열
#
# df1=subset(df, accident==1)
#
# year=df1 %>%
#   group_by(year) %>%
#   summarize(count=n())
#
# ggplot(data = year,
#        mapping = aes(x = year, y = count, col="red",group=1)) + theme_minimal() +
#   geom_point(data = year,size=3) +
#   geom_line(data = year,size=1) +
#   ggtitle("연도별 해양 사고 발생 추이")+theme(text = element_text(size = 20))
```