Pract5

EMD-Marc, Jorge, Alejandro

May 2025

1. One dimensional Partial Dependence Plot

The partial dependence plot shows the marginal effect of a feature on the predicted outcome of a previously fit model.

EXERCISE: Apply PDP to the regression example of predicting bike rentals. Fit a random forest approximation for the prediction of bike rentals (cnt). Use the partial dependence plot to visualize the relationships the model learned. Use the slides shown in class as model.

QUESTION: Analyse the influence of days since 2011, temperature, humidity and wind speed on the predicted bike counts.

```r
# Librerías necesarias
library(randomForest)
```

```
## randomForest 4.7-1.2

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

```r
library(ggplot2)
```

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

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

```r
library(dplyr)
```

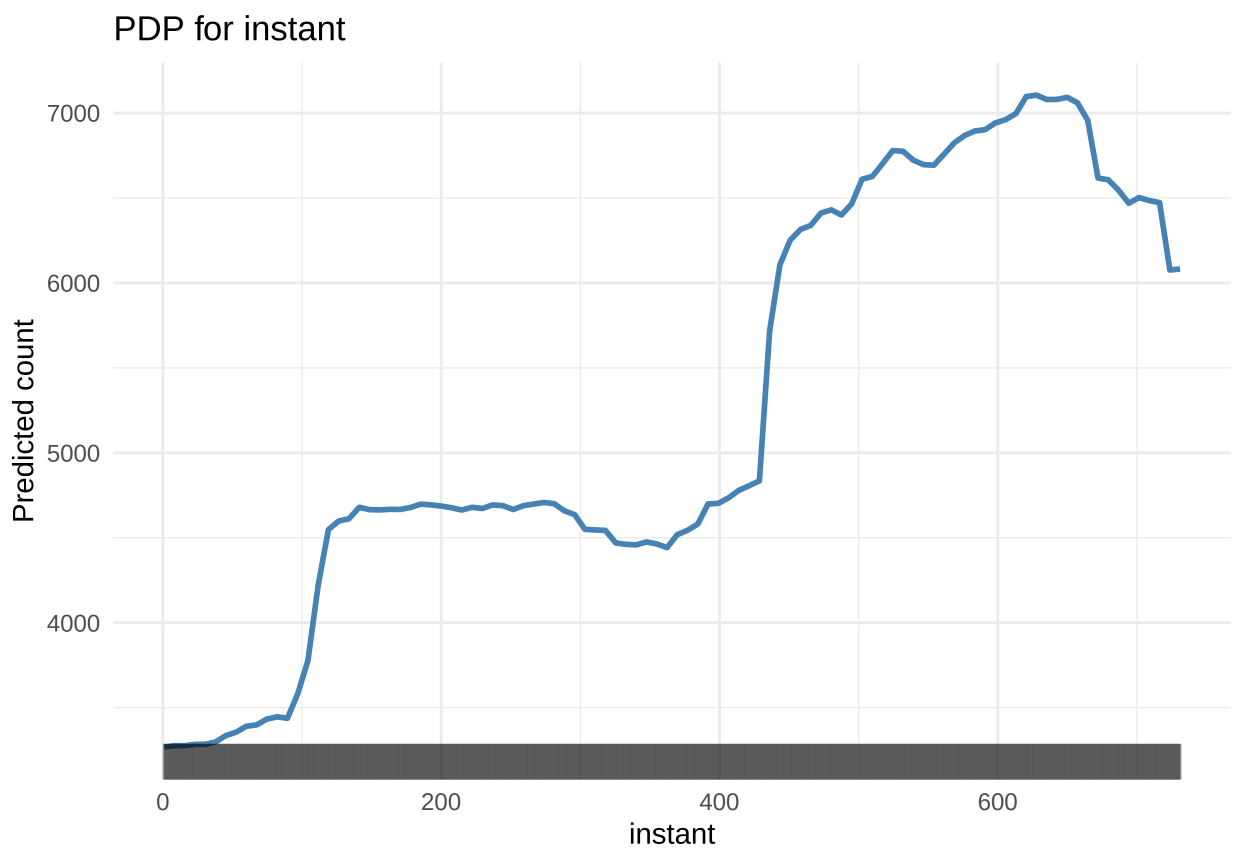
```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:randomForest':
##
##     combine

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

```
# Cargar datos
bike_data <- read.csv("data/day.csv")

# Variables relevantes
bike_data_model <- bike_data %>%
  select(cnt, instant, temp, hum, windspeed)

# Entrenar modelo Random Forest
set.seed(123)
rf_model <- randomForest(cnt ~ ., data = bike_data_model, ntree = 300)

# PDP manual para una variable con rug lines
pdp_manual <- function(varname, data, model, grid.res = 100) {
  # Secuencia de valores para la variable objetivo
  var_seq <- seq(min(data[[varname]]), max(data[[varname]]), length.out = grid.res)

  # Crear nuevo data frame replicando valores promedio y cambiando solo la variable
  pdp_data <- data.frame(matrix(nrow = grid.res, ncol = ncol(data) - 1))
  colnames(pdp_data) <- setdiff(names(data), "cnt")

  for (v in colnames(pdp_data)) {
    pdp_data[[v]] <- if (v == varname) var_seq else mean(data[[v]])
  }

  # Predecir
  preds <- predict(model, newdata = pdp_data)
  pdp_df <- data.frame(x = var_seq, y = preds)

  # Devolver plot ggplot con rug lines
  ggplot(pdp_df, aes(x = x, y = y)) +
    geom_line(color = "steelblue", size = 1) +
    geom_rug(data = data, aes_string(x = varname), inherit.aes = FALSE,
             sides = "b", alpha = 0.4, length = unit(0.05, "npc")) +
    labs(x = varname, y = "Predicted count",
         title = paste("PDP for", varname)) +
    theme_minimal()
}

# Generar los PDPs
p1 <- pdp_manual("instant", bike_data_model, rf_model)
```

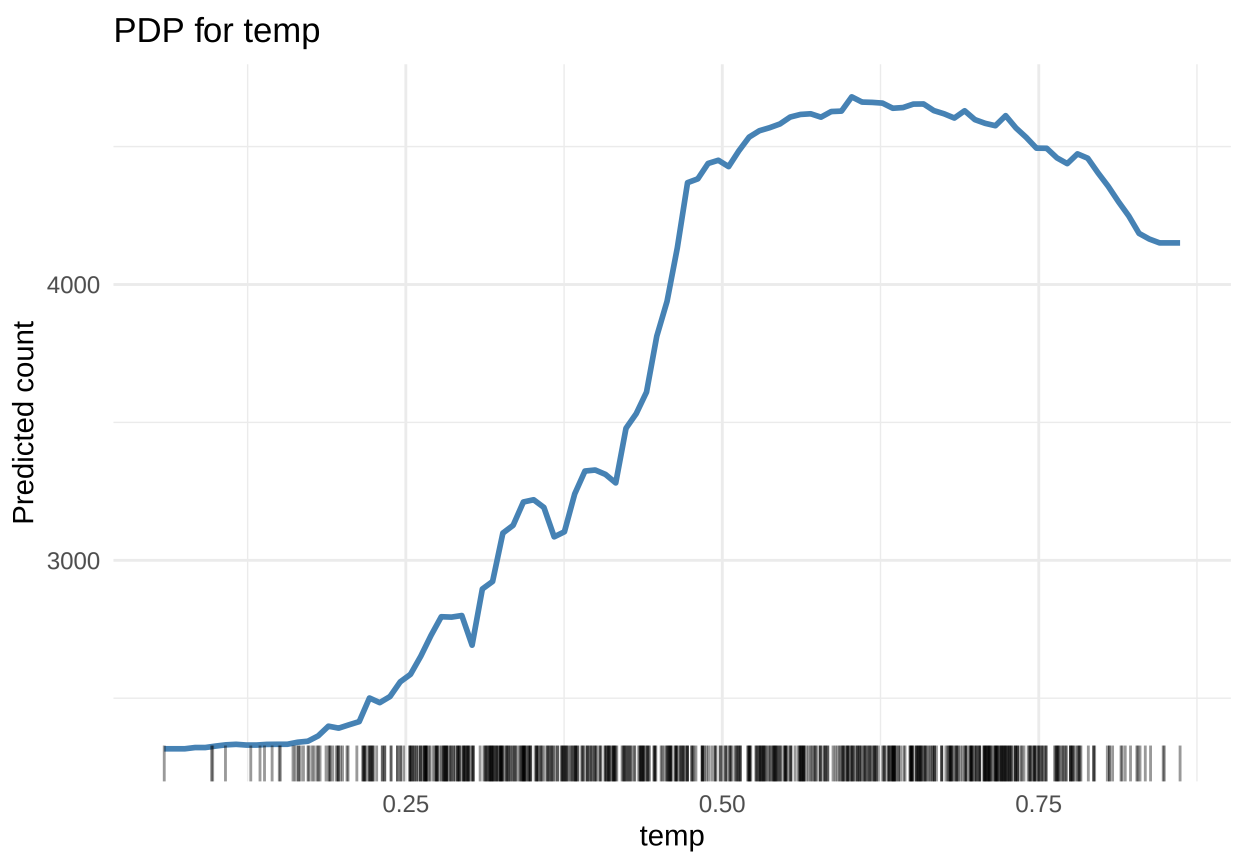
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also 'vignette("ggplot2-in-packages")' for more information.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

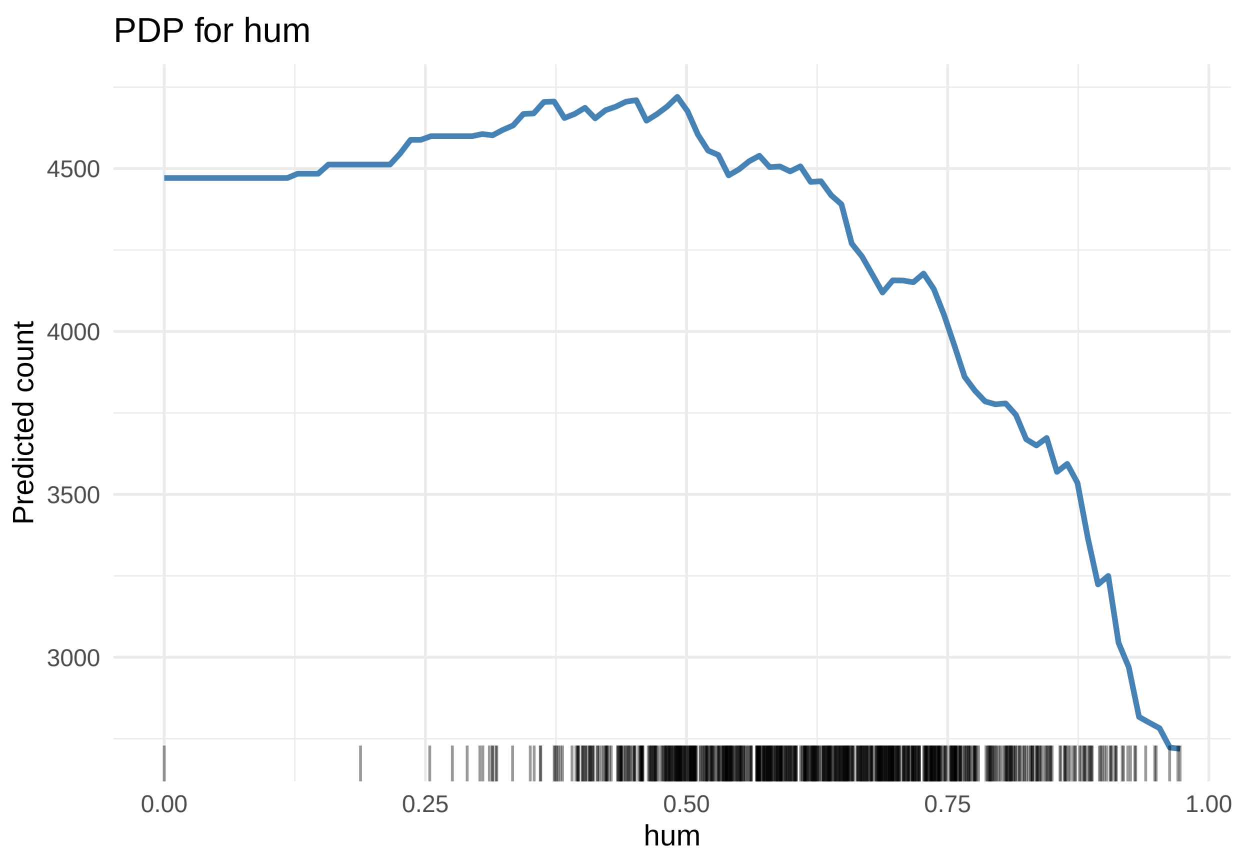
```
p2 <- pdp_manual("temp", bike_data_model, rf_model)
p3 <- pdp_manual("hum", bike_data_model, rf_model)
p4 <- pdp_manual("windspeed", bike_data_model, rf_model)

# Mostrar
print(p1)
```



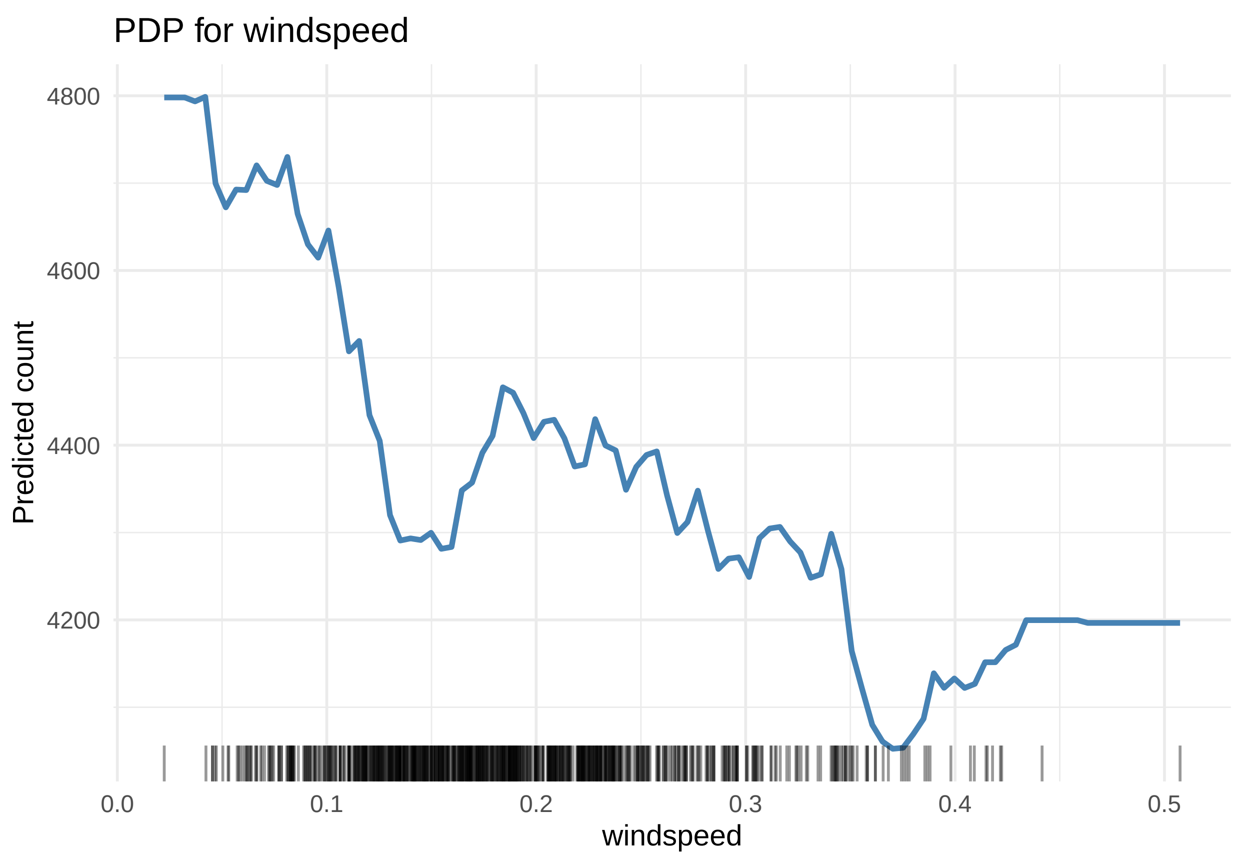
```
print(p2)
```



```r
print(p3)
```


PDP for hum
Predicted count
4500
4000
3500
3000
0.00
0.25
0.50
0.75
1.00
hum

```r
print(p4)
```



2.- Bidimensional Partial Dependency Plot

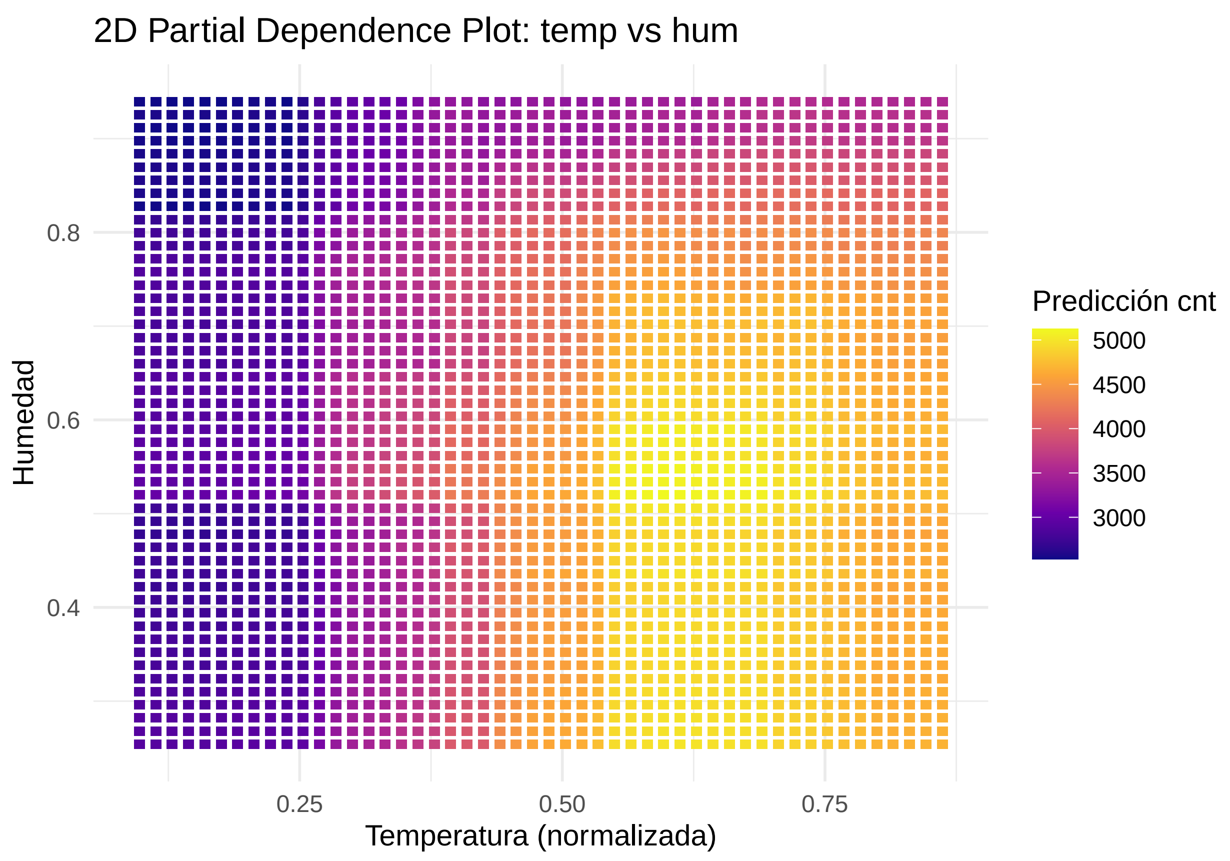
EXERCISE: Generate a 2D Partial Dependency Plot with humidity and temperature to predict the number of bikes rented depending on those parameters.

BE CAREFUL: due to the size, extract a set of random samples from the BBDD before generating the data for the Partial Dependency Plot.

Show the density distribution of both input features with the 2D plot as shown in the class slides.

TIP: Use geom_tile() to generate the 2D plot. Set width and height to avoid holes.

QUESTION: Interpret the results.



3.- PDP to explain the price of a house.

EXERCISE: Apply the previous concepts to predict the price of a house from the database kc_house_data.csv. In this case, use again a random forest approximation for the prediction based on the features bedrooms, bathrooms, sqft_living, sqft_lot, floors and yr_built.

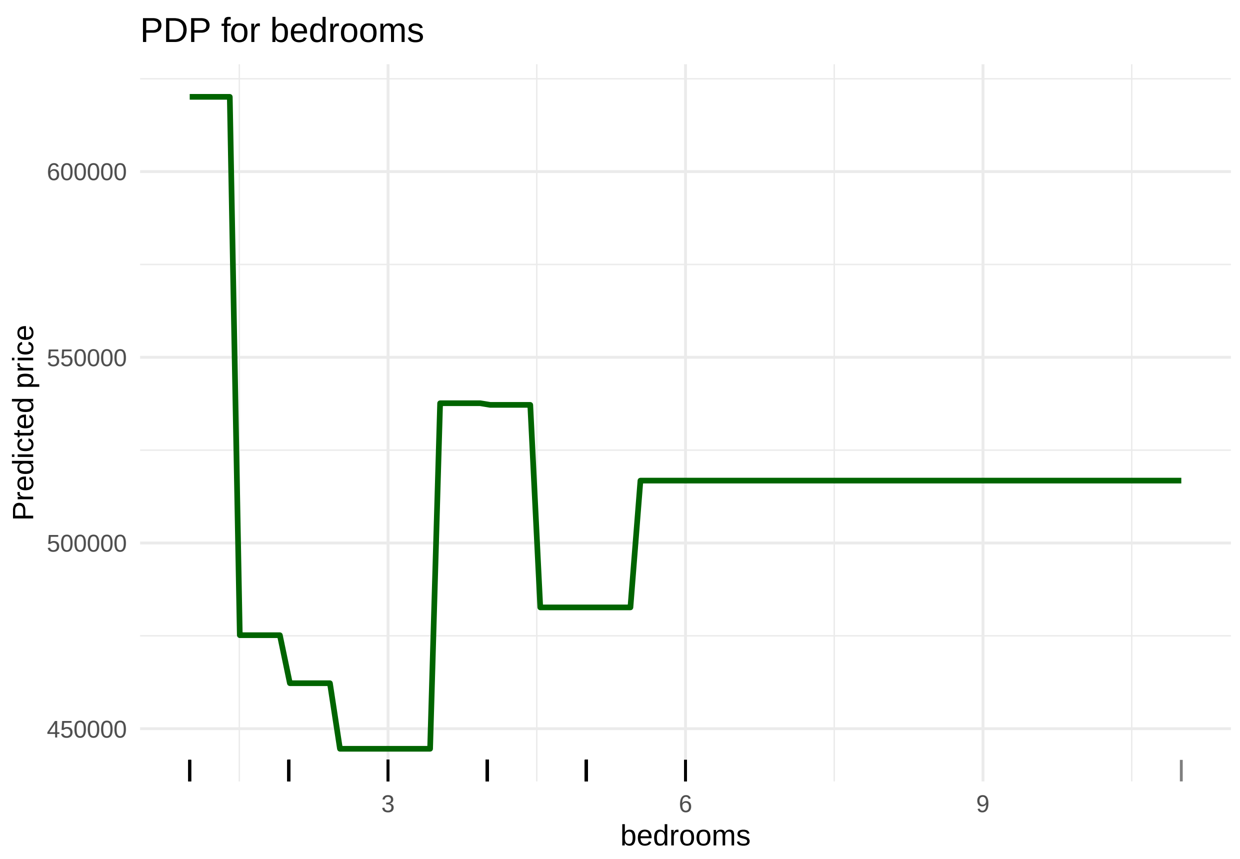
Use the partial dependence plot to visualize the relationships the model learned.

BE CAREFUL: due to the size, extract a set of random samples from the BBDD before generating the data for the Partial Dependency Plot.

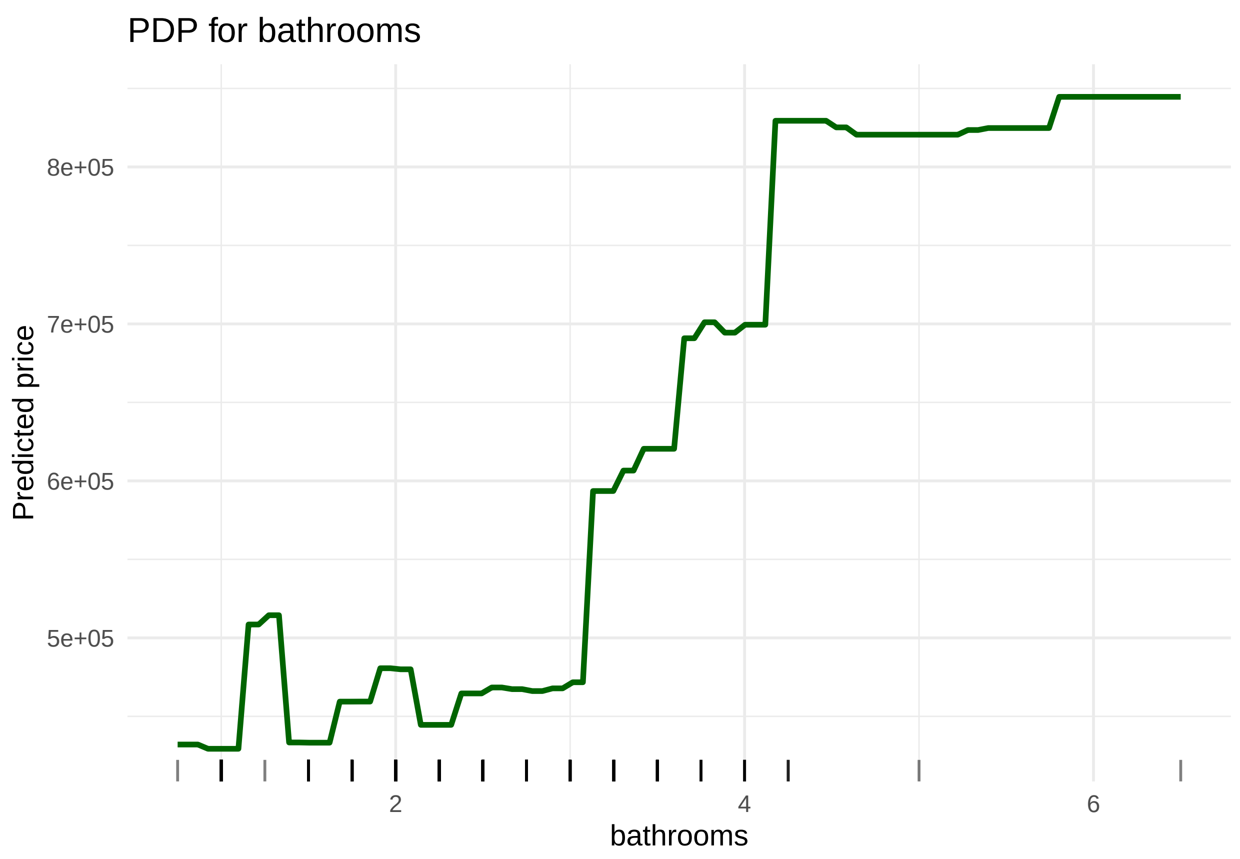
QUESTION: Analyse the influence of bedrooms, bathrooms, sqft_living and floors on the predicted price.

```r
# Cargar librerías necesarias
library(randomForest)
library(ggplot2)
library(dplyr)

# 1. Cargar y preparar datos
kc_data <- read.csv("data/kc_house_data.csv")

# Submuestreo aleatorio por tamaño
set.seed(42)
kc_sample <- kc_data %>%
  select(price, bedrooms, bathrooms, sqft_living, sqft_lot, floors, yr_built) %>%
  sample_n(500)  # ajustar según memoria
```

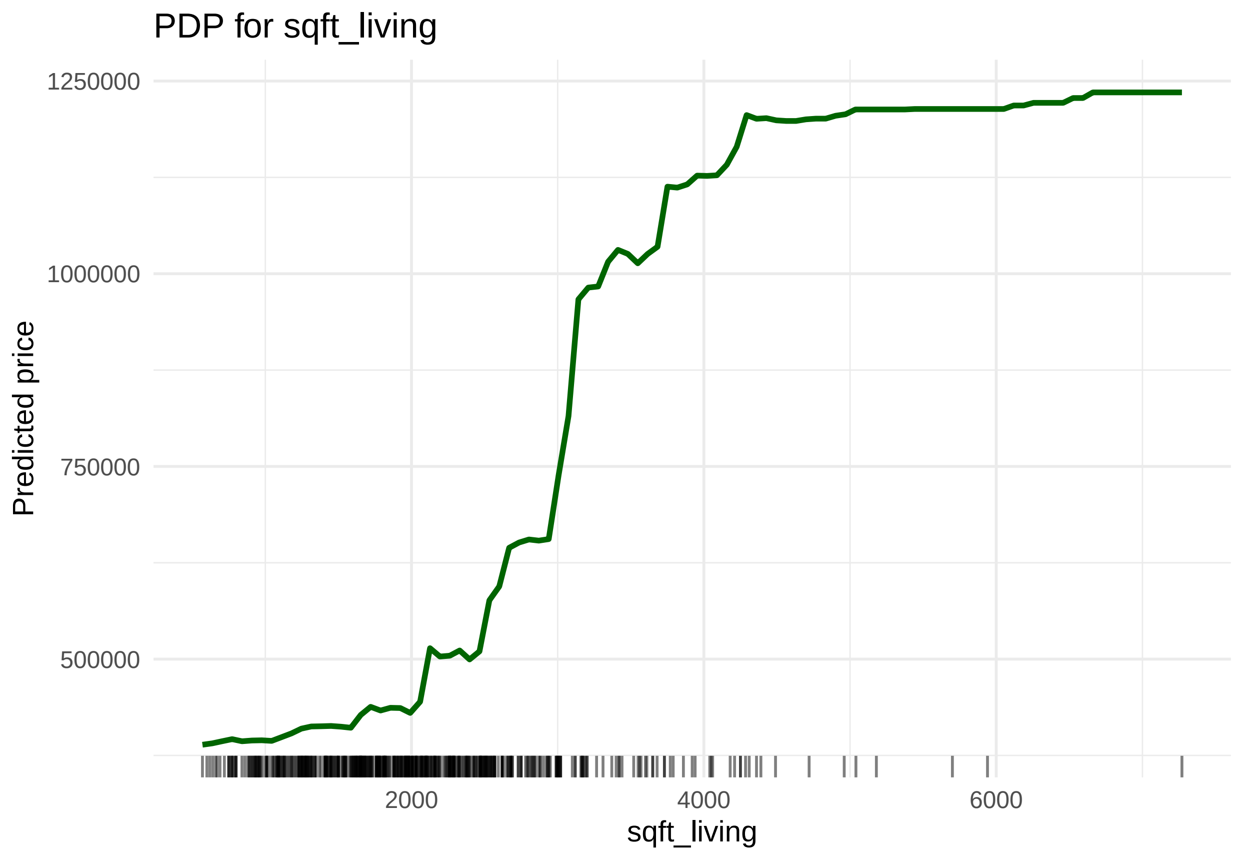
```r
# 2. Entrenar modelo Random Forest
rf_model <- randomForest(price ~ ., data = kc_sample, ntree = 300)

# 3. Función PDP manual con rug plot
pdp_manual <- function(varname, data, model, grid.res = 100) {
  var_seq <- seq(min(data[[varname]]), max(data[[varname]]), length.out = grid.res)
  pdp_data <- data.frame(matrix(nrow = grid.res, ncol = ncol(data) - 1))
  colnames(pdp_data) <- setdiff(names(data), "price")

  for (v in colnames(pdp_data)) {
    pdp_data[[v]] <- if (v == varname) var_seq else mean(data[[v]])
  }

  preds <- predict(model, newdata = pdp_data)
  pdp_df <- data.frame(x = var_seq, y = preds)

  ggplot(pdp_df, aes(x = x, y = y)) +
    geom_line(color = "darkgreen", size = 1) +
    geom_rug(data = data, aes_string(x = varname), inherit.aes = FALSE, sides = "b", alpha = 0.5) +
    labs(x = varname, y = "Predicted price",
         title = paste("PDP for", varname)) +
    theme_minimal()
}

# 4. Generar y mostrar PDPs
p1 <- pdp_manual("bedrooms", kc_sample, rf_model)
p2 <- pdp_manual("bathrooms", kc_sample, rf_model)
p3 <- pdp_manual("sqft_living", kc_sample, rf_model)
p4 <- pdp_manual("floors", kc_sample, rf_model)

print(p1)
```


PDP for bedrooms
Predicted price
600000
550000
500000
450000
3
6
9
bedrooms

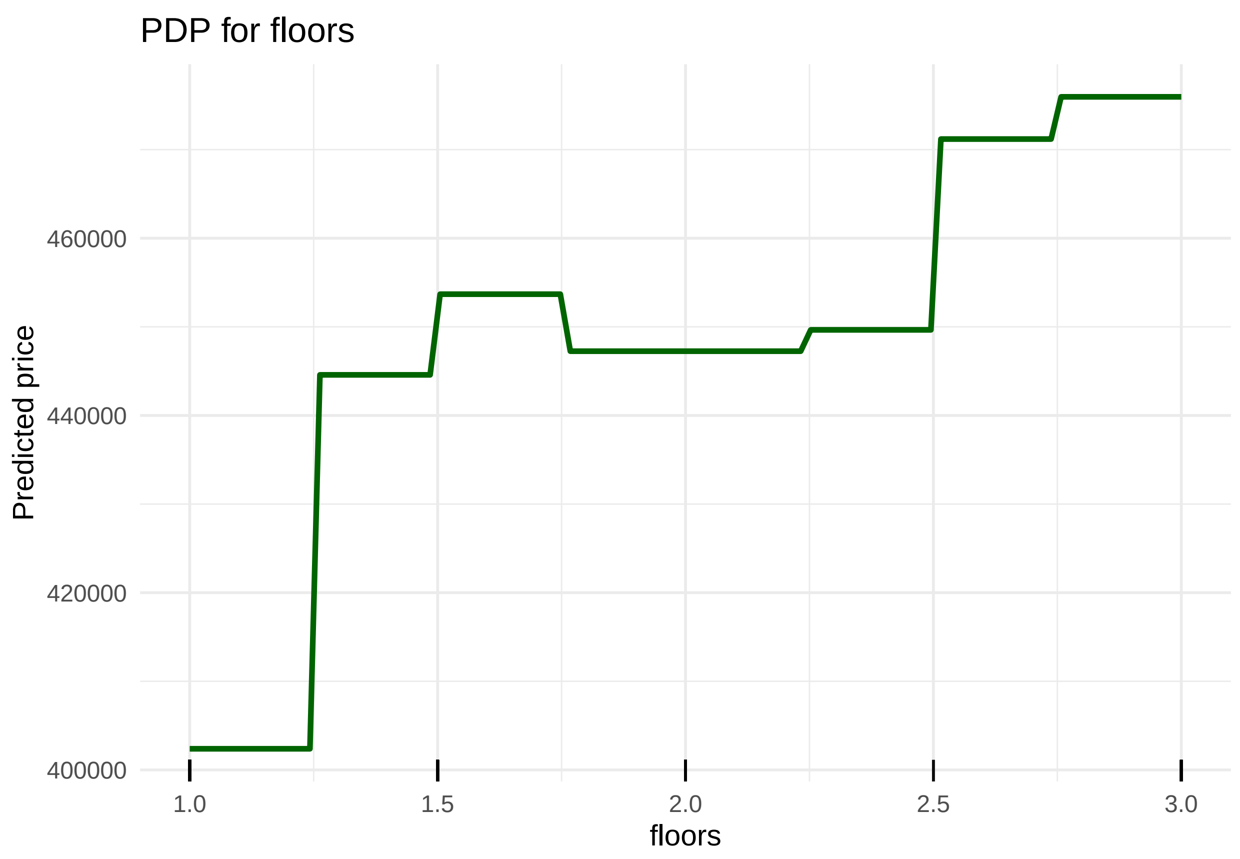
```r
print(p2)
```


PDP for bathrooms
Predicted price
8e+05
7e+05
6e+05
5e+05
2
4
6
bathrooms

```r
print(p3)
```


PDP for sqft_living
Predicted price
1250000
1000000
750000
500000
2000
4000
6000
sqft_living

```r
print(p4)
```



PDP for floors
Predicted price
400000
420000
440000
460000
1.0
1.5
2.0
2.5
3.0
floors