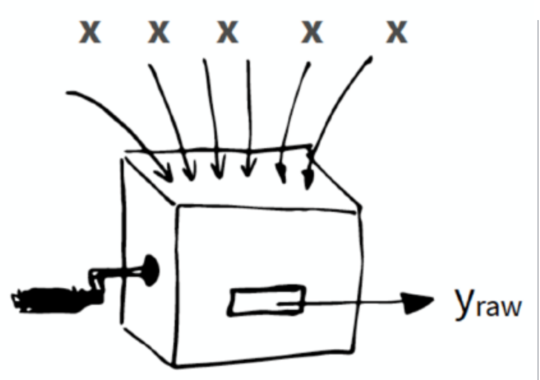
ceterisParibus : : an R package for model agnostic visual exploration

A black box model is a complex predictive model like neural network, random forest or gradient boosting.



Such models are widely used because they are easy to build and often have high accuracy.

But there is a price to pay. Complex models are hard to understand.

It is hard to trust the model if we cannot answer easy questions like:

- What would be the model response for a slightly different input?
- Is this model well fitted for a particular observation?
- How particular variables are linked with the model response?

The **ceterisParibus** package answers such questions.

It is based on an uniform grammar for model agnostic visual explanations.

Basic of Ceteris Paribus Profiles

Ceteris paribus is a Latin phrase meaning *all else unchanged*. This principle shows how the model response depends on *changes in a single input variable*, keeping *all other variables unchanged*.

Fusion of such profiles calculated for different observations, variables, models help to understand how a black model is working.

Two main function in the **ceterisParibus** package are:

- `ceteris_paribus()` calculates profiles for selected models and observations.
- `plot()` plots selected set of profiles.

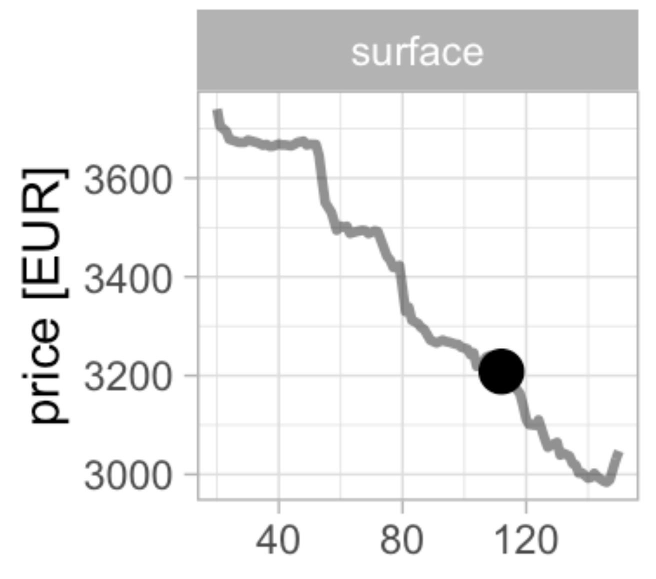
Models needs to be preprocessed with the `explain()` function from **DALEX** package.

```
library("DALEX")
library("ceterisParibus")
library("randomForest")

rf_model <- randomForest(m2.price ~ year + surface +
              floor + district, data = apartments)

explainer_rf <- explain(rf_model,
                   data = apartments[,2:6],
                   y = apartments$m2.price)
```

Single Model Response

For a single observation the CP profile shows how the model response would change if only a single variable is changed. It's useful for question like: what would happen if variable X = x

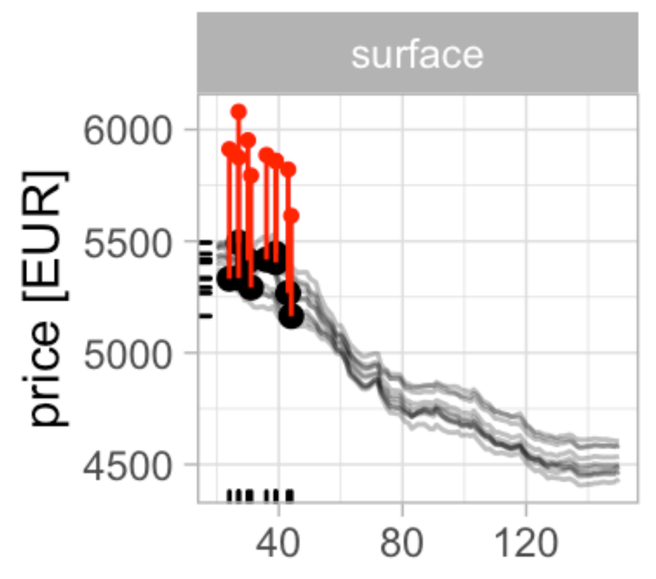
```
new_obs <- apartments[1,]
cp_rf <- ceteris_paribus(explainer_rf,
               new_obs, y = new_obs$m2.price)
plot(cp_rf, selected_variables = "surface")
```



Local Model Fit

CP profiles for observations similar to the one of interest show how stable is the model response. Since we know the true values for these points, residuals show how accurate is the model fit locally.

```
neighbours <- select_neighbours(apartments,
               new_obs, n = 15)
cp_rf <- ceteris_paribus(explainer_rf,
               neighbours, y = neighbours$m2.price)
plot(cp_rf, show_residuals = TRUE,
               selected_variables = "surface")
```

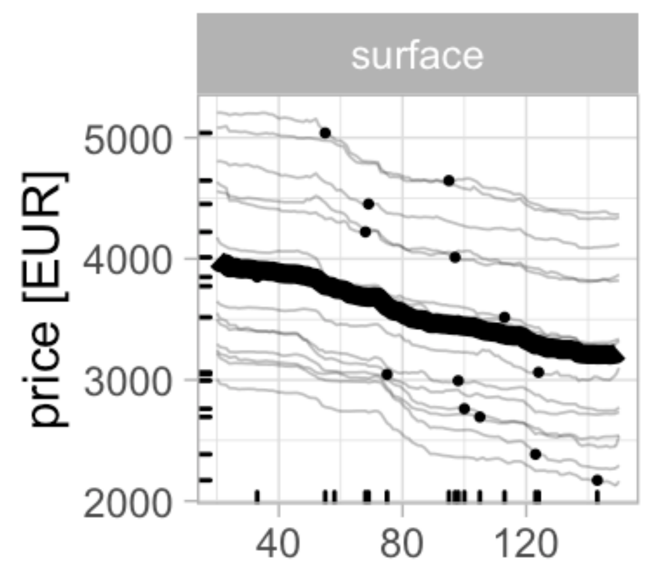


Average Model Response

Partial Dependency Plots show an average from profiles for all observations. It is a useful tool to summarise global relation between a variable and model response.

In such context CP profiles are sometimes named *Individual Conditional Expectations (ICE).*

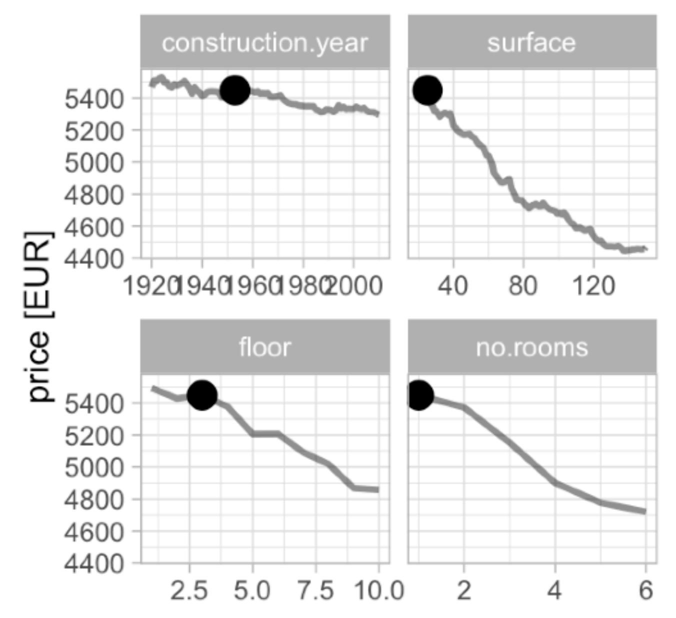
```
cp_rf <- ceteris_paribus(explainer_rf,
             apartments, y = apartments$m2.price)
plot(cp_rf, selected_variables = "surface",
             aggregate_profiles = mean,
             show_observations = FALSE)
```



Many Variables

It is useful to trace and plot CP profiles for many variables at the same time. This gives more complete picture of the black box model.

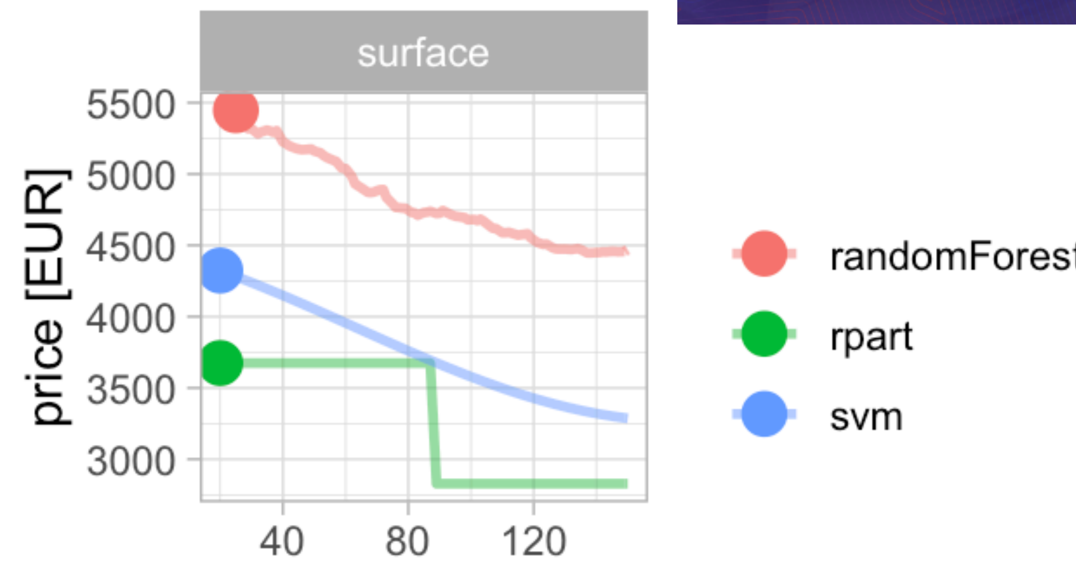
```
plot(cp_rf,
     selected_variables = c("construction.year",
                   "surface", "floor", "np.rooms"))
```



Many Models

It is useful to cross compare different models for one observation, few observations, or an average. This way we may grasp some differences between models.

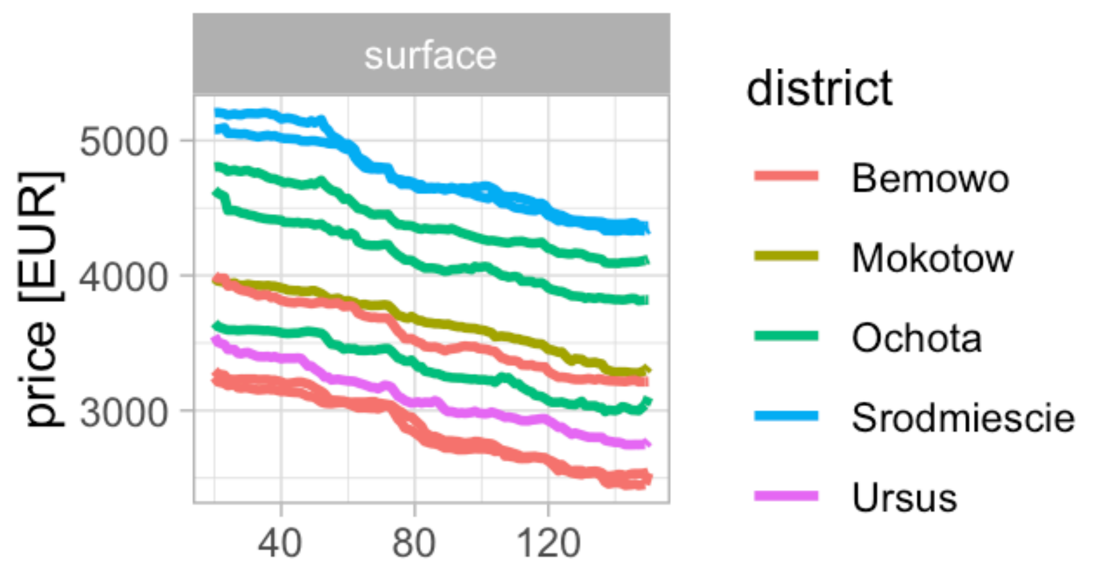
```
library("e1071")
svm_model <- svm(m2.price ~ year + surface +
             floor + district, data = apartments)
explainer_svm <- explain(svm_model,
     data = apartments, y = apartments$m2.price)
cp_svm <- ceteris_paribus(explainer_svm,
                new_obs, y = new_obs$m2.price)
plot(cp_rf, cp_svm,  color = "_label_",
 selected_variables = "surface")
```



Model Interactions

If there are no interactions with a selected variable then all CP profiles are parallel. If CP profiles are not parallel use colors to understand the nature of an interaction.

```
plot(cp_rf,
     color = "district",
     selected_variables =
                  c("surface", "district"))
```

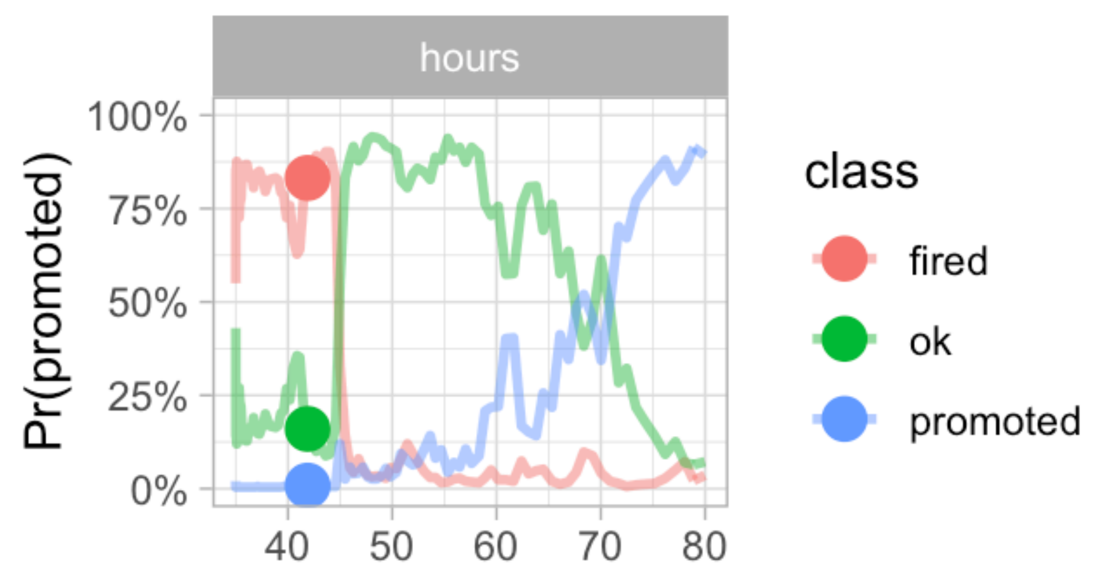


Multiclass Models

Plotting many models in a single plot is especially useful for multiclass models or multivariate models, i.e. models with multidimensional response.

This way a single plot can show how different partial outputs are dependent on variable of interest.

```
plot(cp_rf1, cp_rf2, cp_rf3,
     color = "_label_",
     selected_variables = "hours")
```

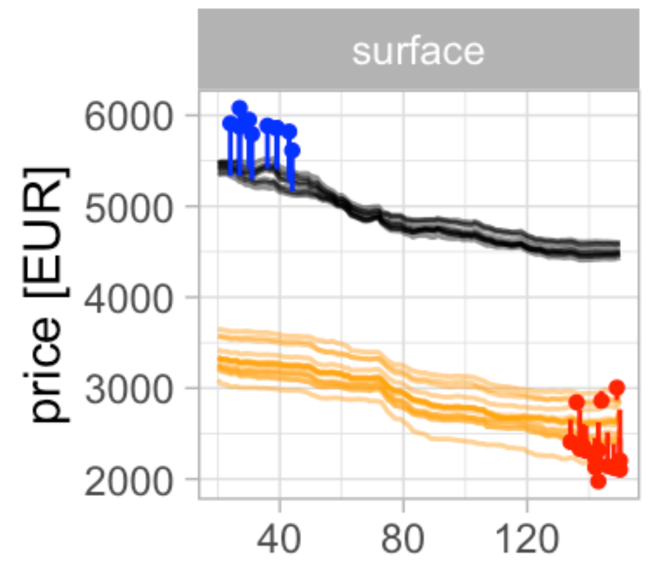


Multiple layers

Multiple Ceteris Paribus profiles can be superimposed in a single plot. New layers can be added with the **ceteris_paribus_layer()** function.

This gives a lot of flexibility. Use with care.

```
plot(cp_rf_A,
     color_residuals = "blue",
     selected_variables = "surface") +
  ceteris_paribus_layer(cp_rf_B,
          color_residuals = "red",
          selected_variables = "surface")
```


surface
price [EUR]