The partial dependence plot is a nice tool to analyse the impact of some explanatory variables when using nonlinear models, such as a random forest, or some gradient boosting.The idea (in dimension 2), given a model $m(x_1,x_2)$ for $\mathbb{E}[Y|X_1=x_1,X_2=x_2]$. The partial dependence plot for variable x_1 is model m is function p_1 defined as $x_1\mapsto \mathbb{E}_{\mathbb{P}_{X_2}}[m(x_1,X_2)]$. This can be approximated, using some dataset using $\widehat{p}_1(x_1)=\frac{1}{n}\sum_{i=1}^n m(x_1,x_{2,i})$My concern here what the interpretation of that plot when there are some (strongly) correlated covariates. Let us generate some dataset to start with

```
n=1000
library(mnormt)
r=.7
set.seed(1234)
X = rmnorm(n,mean = c(0,0),varcov = matrix(c(1,r,r,1),2,2))
Y = 1+X[,1]-2*X[,2]+rnorm(n)/2
df = data.frame(Y=Y,X1=X[,1],X2=X[,2])
```

As we can see, the true model is here is $y_i=\beta_0+\beta_1 x_{1,i}+\beta_2x_{2,i}+\varepsilon_i$ where $\beta_1 =1$ but the two variables are positively correlated, and the second one has a strong negative impact. Note that here

```
reg = lm(Y~.,data=df)
summary(reg)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.01414    0.01601   63.35   <2e-16 ***
X1           1.02268    0.02305   44.37   <2e-16 ***
X2          -2.03248    0.02342  -86.80   <2e-16 ***
```

If we estimate a wrongly specified model $y_i=b_0+b_1 x_{1,i}+\eta_i$, we would get

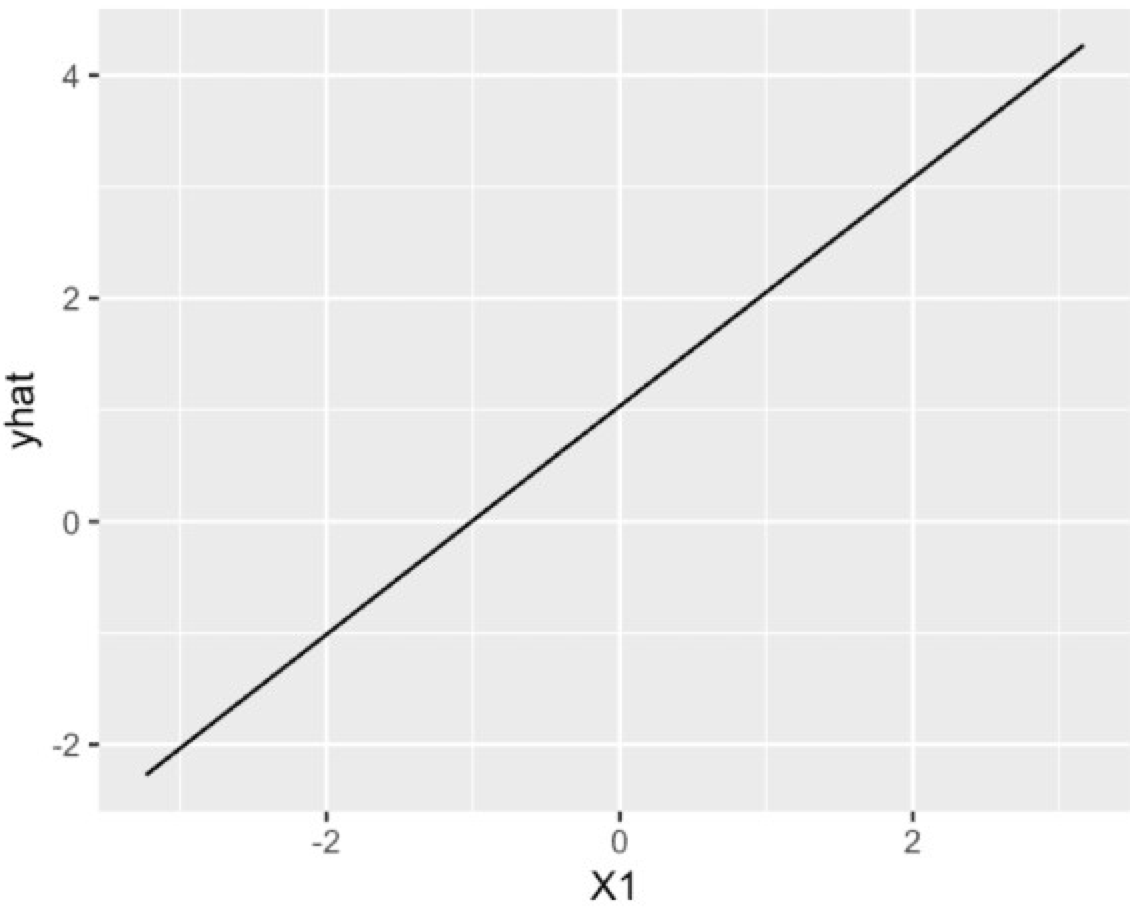
```
reg1 = lm(Y~X1,data=df)
summary(reg1)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.03522    0.04680  22.121   <2e-16 ***
X1          -0.44148    0.04591  -9.616   <2e-16 ***
```

Thus, on the proper model, $\widehat{\beta}_1\sim+1.02$ while $\widehat{b}_1\sim-0.44$ on the mispecified model.

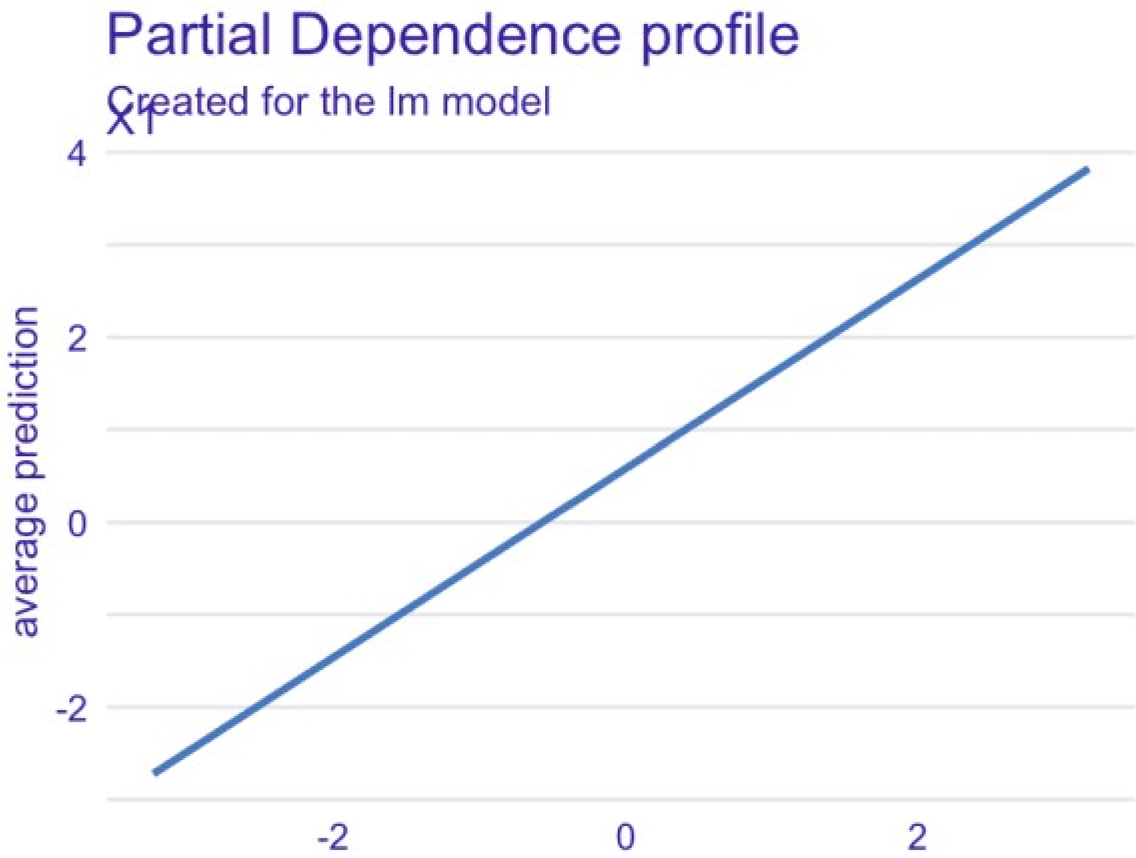
Now, let us look at the parial dependence plot of the good model, using standard R dedicated packages,

```
library(pdp)
pdp::partial(reg, pred.var = "X1", plot = TRUE,
              plot.engine = "ggplot2")
```



which is the linear line \(y=1+x\), that corresponds to \(y=\beta_0+\beta_1x\).

```
library(DALEX)
plot(DALEX::single_variable(DALEX::explain(reg,
data=df),variable = "X1",type = "pdp"))
```

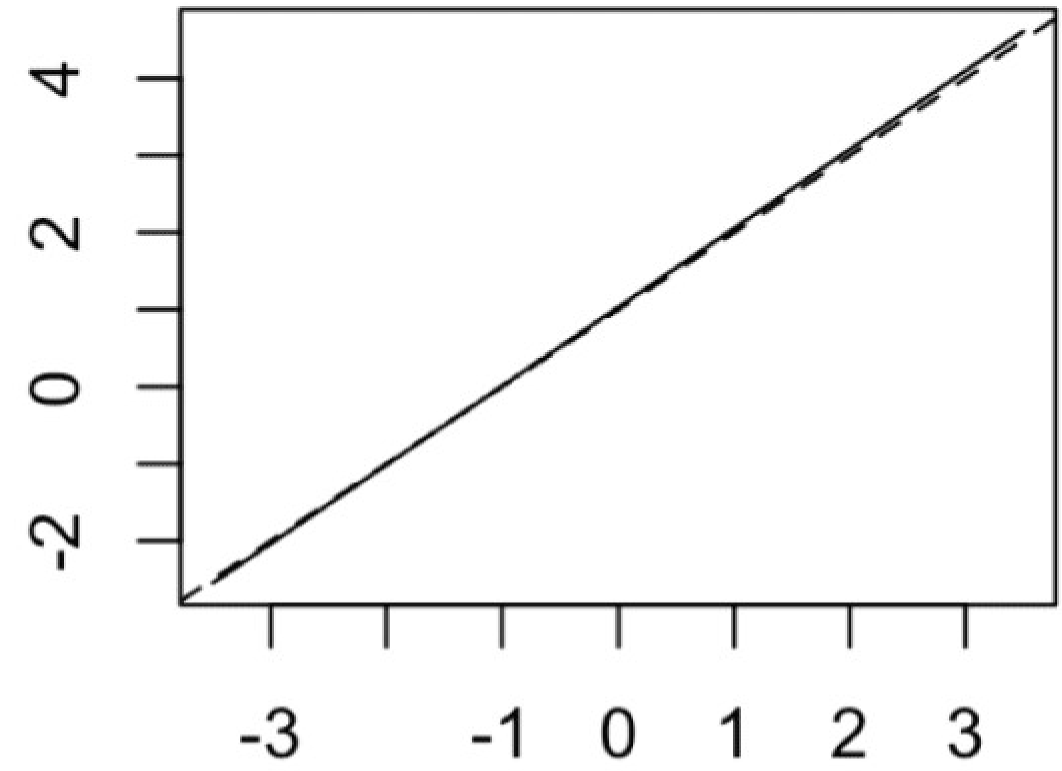


which corresponds to the previous graph. Here, it is also possible to creaste our own function to compute that partial dependence plot,

```
pdp1 = function(x1){
  nd = data.frame(X1=x1,X2=df$X2)
  mean(predict(reg,newdata=nd))
}
```

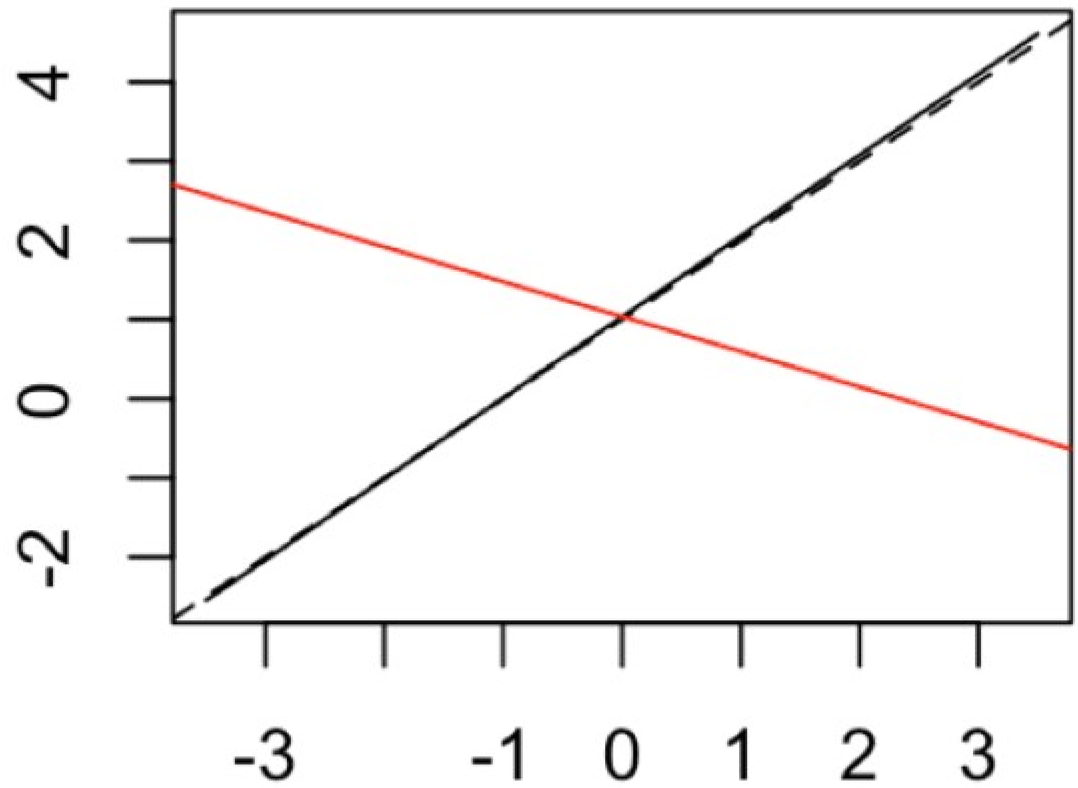
that will be the straight line below (the dotted line is the theoretical one \(y=1+x\),

```
vx=seq(-3.5,3.5,length=101)
vpdp1 = Vectorize(pdp1)(vx)
plot(vx,vpdp1,type="l")
abline(a=1,b=1,lty=2)
```



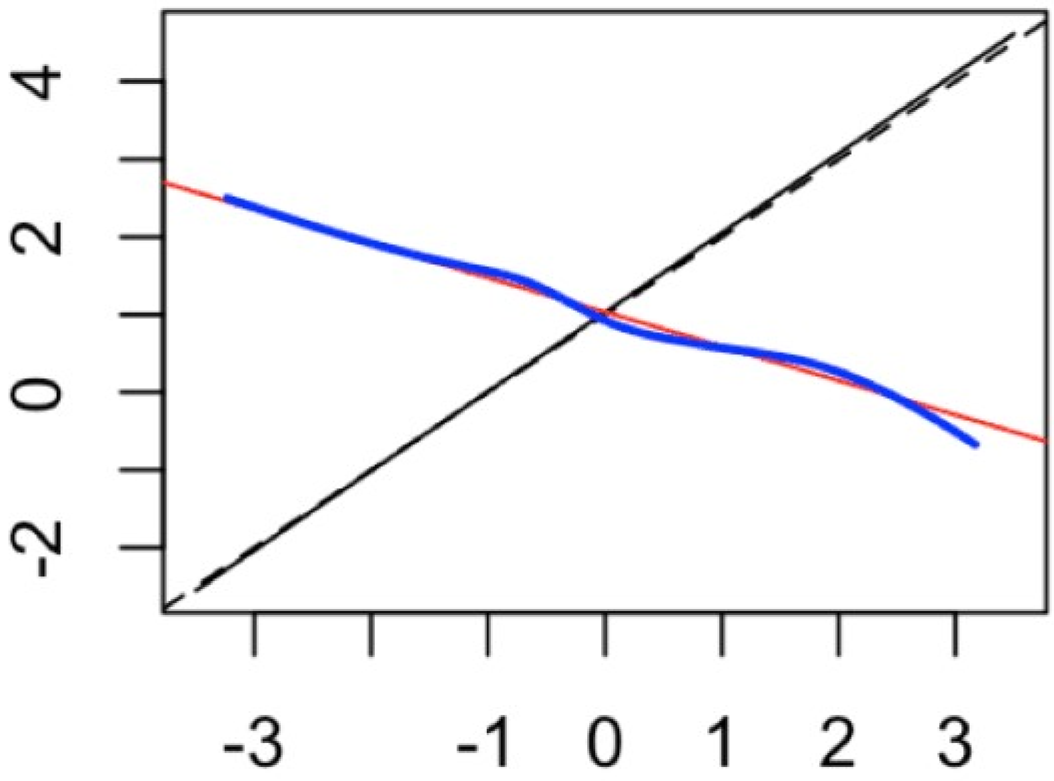
which is very different from the univariate regression on \(x_1\)

```
abline(reg1,col="red")
```



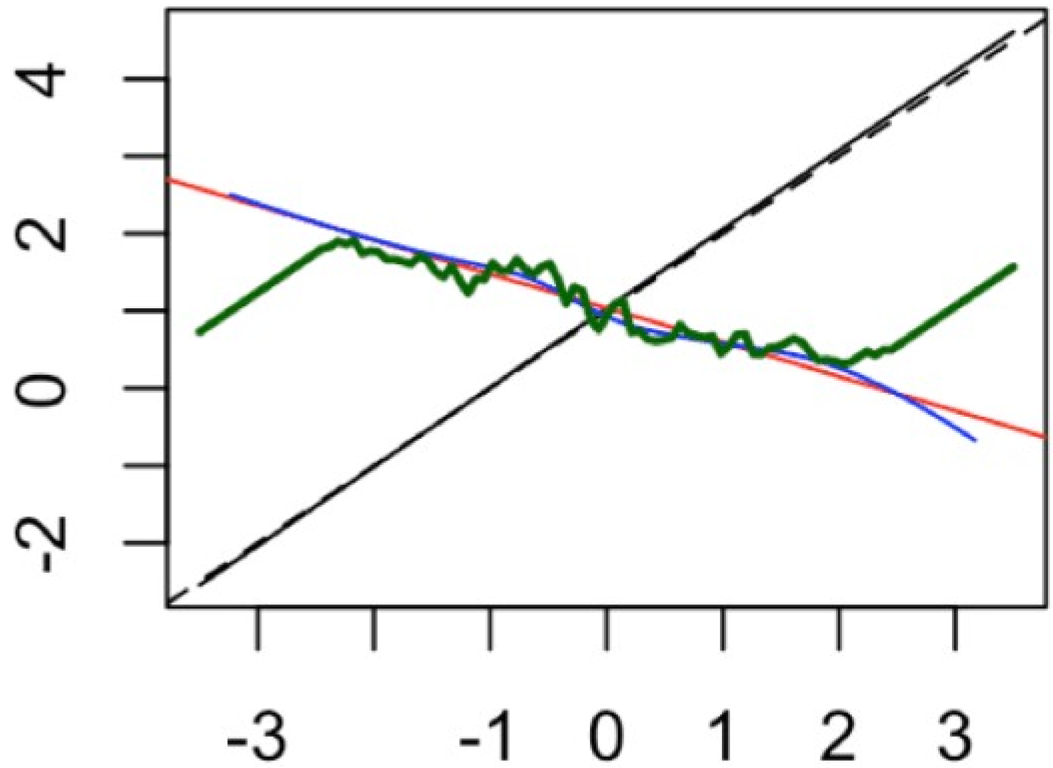
Actually, the later is very consistent with a local regression, only on \(x_1\)

```
library(locfit)
lines(locfit(Y~X1,data=df),col="blue")
```



Now, to get back to the definition of the partial dependence plot, $x_1\mapsto\mathbb{E}_{\mathbb{P}_{X_2}}[m(x_1,X_2)]$, in the context of correlated variable, I was wondering if it would not make more sense to consider some local version actually, something like $x_1\mapsto\mathbb{E}_{\mathbb{P}_{X_2|X_1}}[m(x_1,X_2)]$. My intuition was that, somehow, it did not make any sense to consider any X_2 while X_1 was fixed (and equal to x_1). But it would make more sense actually to look at more valid X_2's given the value of X_1. And a natural estimate could be some k neareast-neighbors, i.e. $\tilde{p}_1(x_1)=\frac{1}{k}\sum_{i\in\mathcal{V}_k(x)}^n m(x_1,x_{2,i})$ where $\mathcal{V}_k(x)$ is the set of indices of the k x_i's that are the closest to x, i.e.

```
lpdp1 = function(x1){
  nd = data.frame(X1=x1,X2=df$X2)
  idx = rank(abs(df$X1-x1))
  mean(predict(reg,newdata=nd[idx<50,]))
}
vlpdp1 = Vectorize(lpdp1)(vx)
lines(vx,vlpdp1,col="darkgreen",lwd=2)
```



Surprisingly (?), this local partial dependence plot gives a curve that corresponds to the simple regression…