D. Makowski
Université Paris-Saclay
INRAE

Outline

- Definition & main principles
- Several extensions of linear regression
- Trees and forests
- Deep learning

Linear regression

$$Y = X\theta + \varepsilon$$

Y is a *N*-vector of output values,
X is a matrix *N* by *P* of inputs (design),
θ is a *P*-vector of parameters,
ε is a *N*-vector of residuals

- Simple linear regression (including one input only),
- Polynomial regression
- Linear model with interactions

$$Y = X\theta + \varepsilon$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{pmatrix} = \begin{array}{c} \begin{array}{cccc} \mathrm{X}_1 & \mathrm{X}_2 & \ldots & \mathrm{X}_\mathrm{P} \end{array} \\ \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1P} \\ x_{21} & x_{22} & \cdots & x_{2P} \\ \cdots & \cdots & \cdots & \cdots \\ x_{N1} & x_{N2} & \cdots & x_{NP} \end{pmatrix} \end{array} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \cdots \\ \theta_P \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \cdots \\ \varepsilon_N \end{pmatrix}$$

$$Y = X\theta + \varepsilon$$

$$\begin{pmatrix} y_1 \\ \cdots \\ y_N \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ \cdots & \cdots \\ 1 & x_N \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \cdots \\ \varepsilon_N \end{pmatrix}$$

Several issues

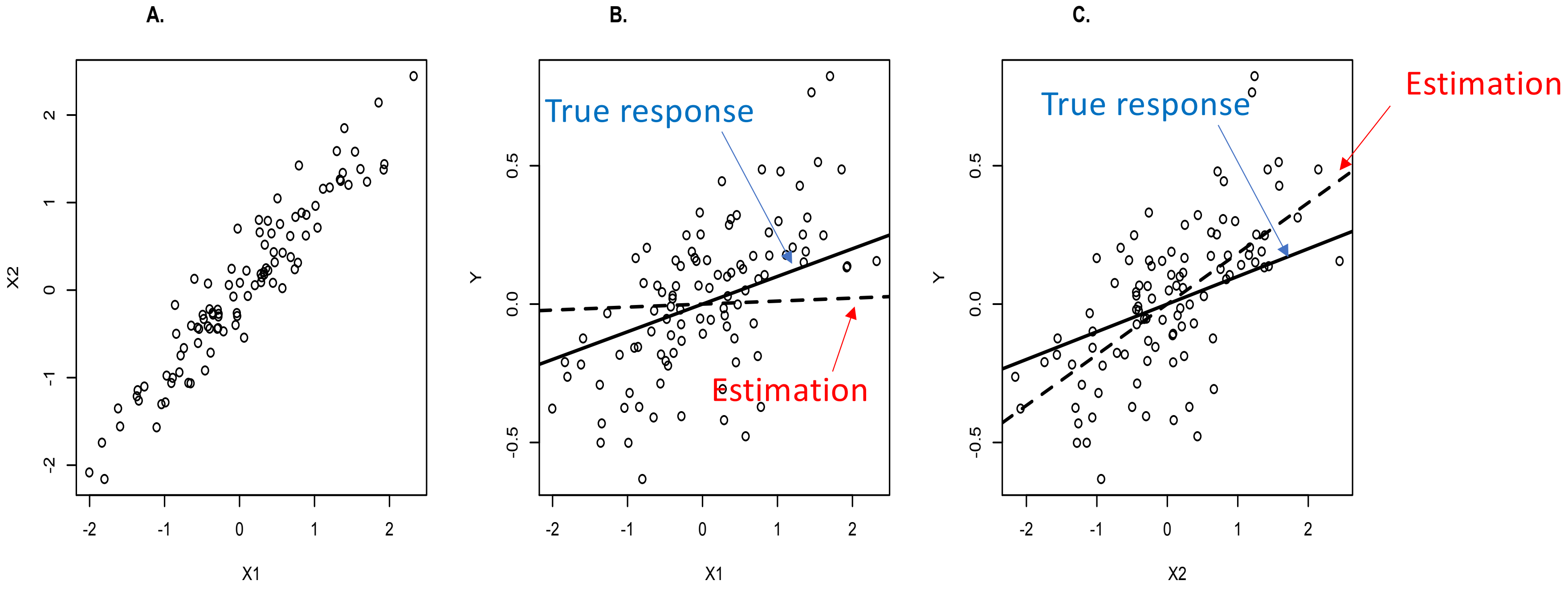
- Inputs X_1, …, X_P are sometimes (strongly) correlated
- Inputs X_1, …, X_P may have non-linear effects (unknown response shape),
- Too many inputs
- Need to estimate extreme responses, not mean response

Several issues

- **Inputs X_1, …, X_P are sometimes (strongly) correlated**
- Inputs X_1, …, X_P may have non-linear effects (unknown response shape),
- Too many inputs
- Need to estimate extreme responses, not mean response

Why can correlated inputs be an issue?

100 data generated with $Y = 0.1X_1 + 0.1X_2 + \varepsilon$



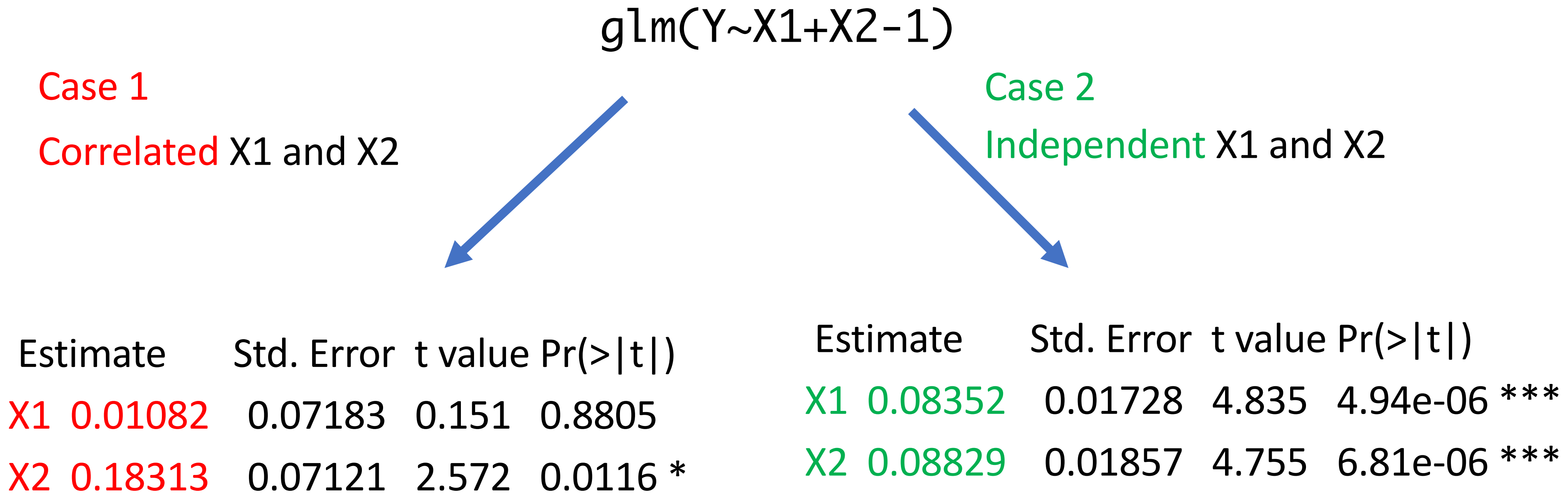
True response

Estimation

True response

Estimation

Correlation between X1 and X2 =0.95



`glm(Y~X1+X2-1)`

Case 1

Correlated X1 and X2

```
    Estimate   Std. Error  t value Pr(>|t|)
X1  0.01082    0.07183   0.151   0.8805
X2  0.18313    0.07121   2.572   0.0116 *
```

Case 2

Independent X1 and X2

```
    Estimate   Std. Error  t value Pr(>|t|)
X1  0.08352    0.01728   4.835   4.94e-06 ***
X2  0.08829    0.01857   4.755   6.81e-06 ***
```

Both results are obtained with 100 data

```
glm(Y~X1+X2-1)
```

Case 1

Correlated X1 and X2

```
    Estimate    Std. Error  t value Pr(>|t|)
X1  0.01082     0.07183     0.151   0.8805
X2  0.18313     0.07121     2.572   0.0116 *
```

Case 2

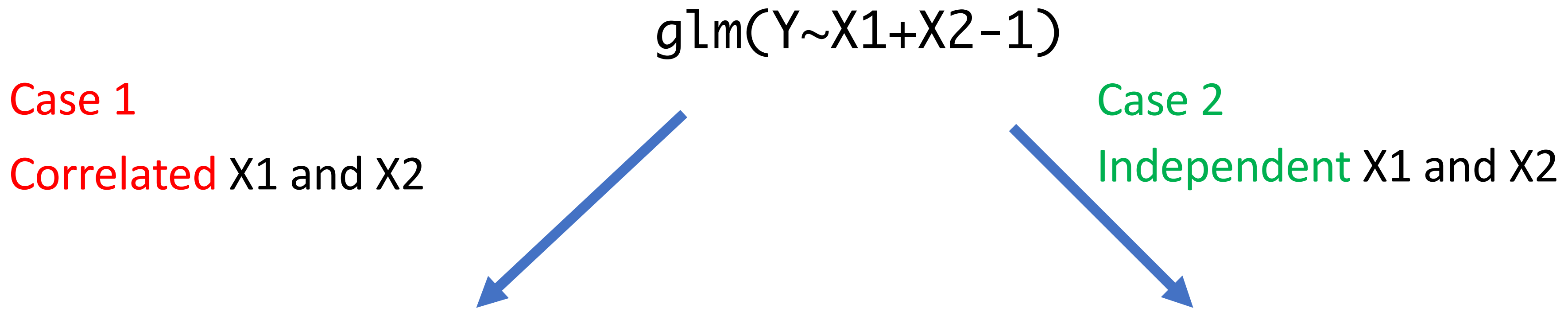
Independent X1 and X2

```
    Estimate    Std. Error  t value Pr(>|t|)
X1  0.08352     0.01728     4.835   4.94e-06 ***
X2  0.08829     0.01857     4.755   6.81e-06 ***
```

Question 1. Effect of X1 significant in cases 1 and 2?

A. Yes

B. No



`glm(Y~X1+X2-1)`

Case 1

Correlated X1 and X2

Case 2

Independent X1 and X2

```
    Estimate    Std. Error  t value Pr(>|t|)
X1  0.01082     0.07183     0.151   0.8805
X2  0.18313     0.07121     2.572   0.0116 *
```

```
    Estimate    Std. Error  t value Pr(>|t|)
X1  0.08352     0.01728     4.835   4.94e-06 ***
X2  0.08829     0.01857     4.755   6.81e-06 ***
```

Question 1. Effect of X1 significant in cases 1 and 2?

A. Yes

B. No

Question 2. Which case shows the most accurate estimated values?

A. Case 1

B. Case 2

Idea: use independent combinations of X_1, X_2, …, X_P

- Principal component regression (PCR)
- Partial least square regression (PLSR)

Principal component analysis

- Replace the initial input variables (X_1, ..., X_P) with new independent variables
- The new variables correspond to linear combinations of the old ones
- These linear combinations are chosen so as to have a maximum variance

$$Y = \alpha_0 + \alpha_1 X_1 + \cdots + \alpha_P X_P + \varepsilon$$

$$Z_1 = \sum_{j=1}^{P} \beta_{1j} X_j \qquad Z_2 = \sum_{j=1}^{P} \beta_{2j} X_j \qquad \text{etc.}$$

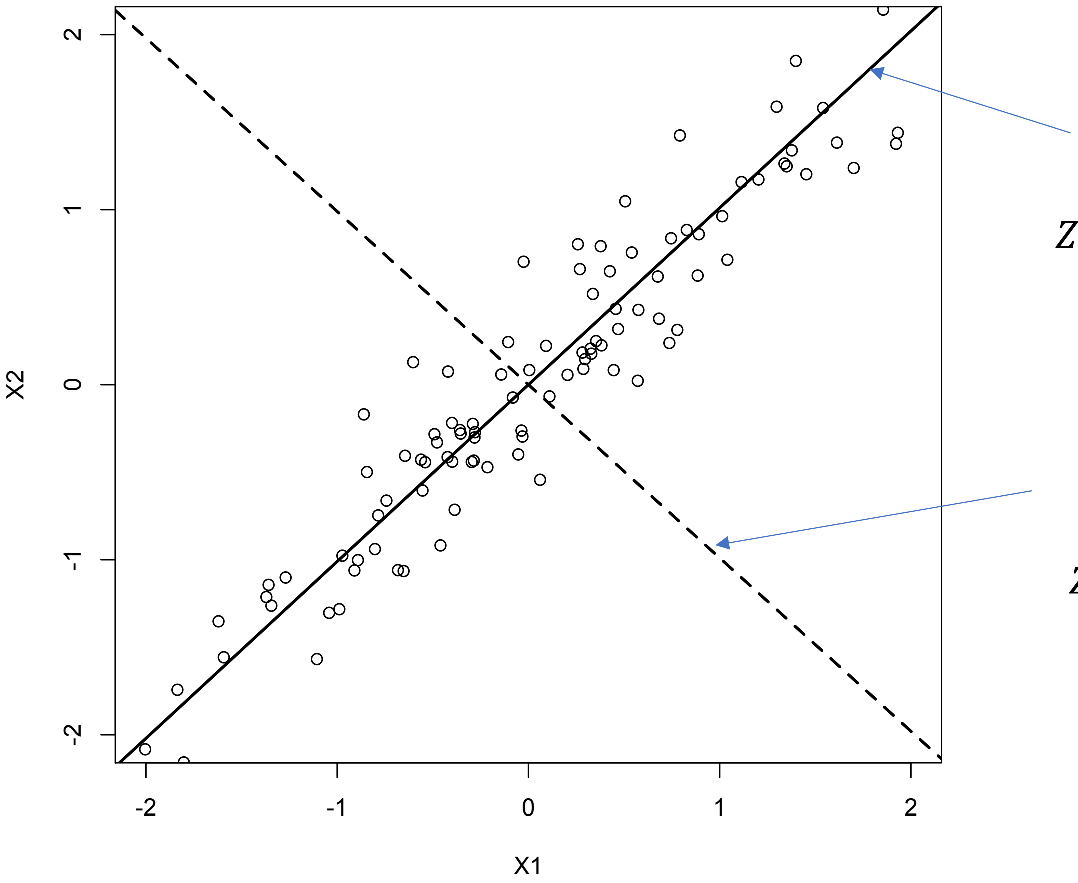
$$Y = \gamma_0 + \gamma_1 Z_1 + \cdots + \gamma_K Z_K + \varepsilon$$

Principal components

The components are calculated from the eigenvectors of the variance-covariance matrix of

$X_1, \ldots, X_P$

→ Diagonalization of the variance-covariance matrix



Component 1

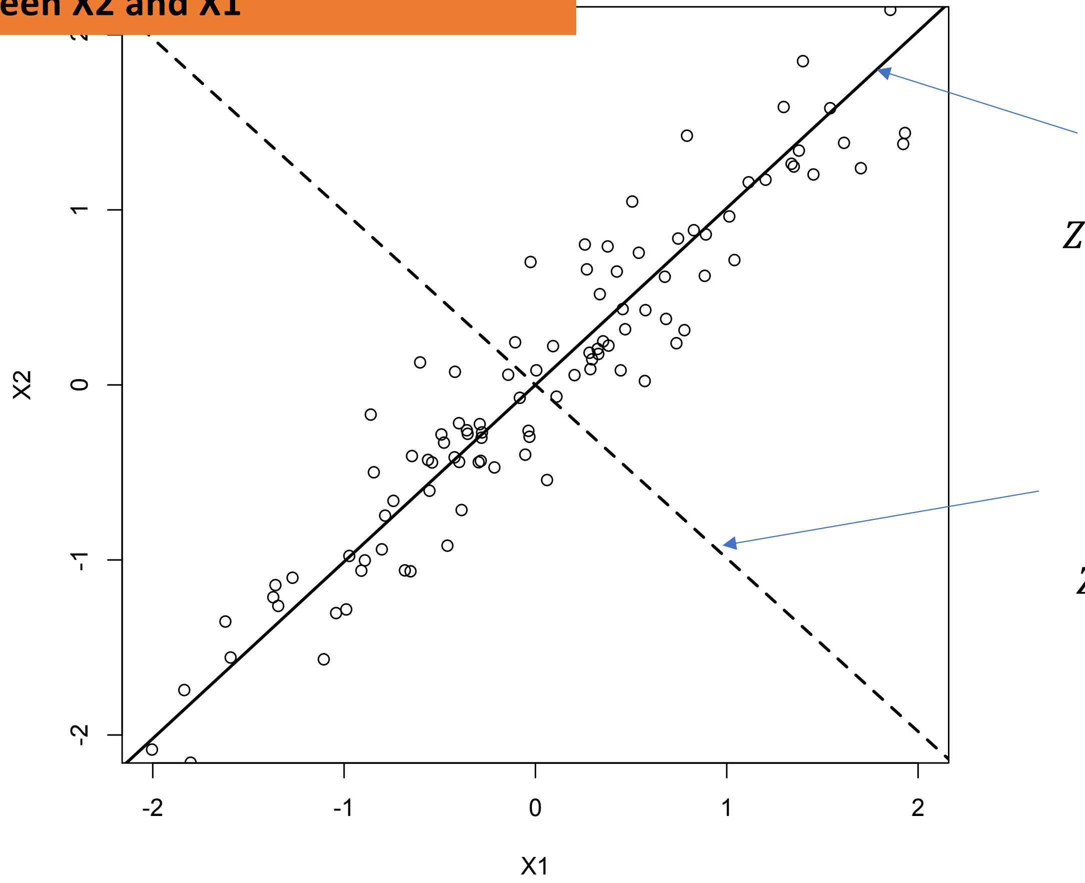
$$Z_1 = 0.704X_1 + 0.711X_2$$

Component 2

$$Z_2 = 0.711X_1 - 0.704X_2$$

Question 1: Component 1 is closer to

A. **The mean of X1 and X2**
B. **The difference between X2 and X1**



Component 1

$$Z_1 = 0.704X_1 + 0.711X_2$$

Component 2

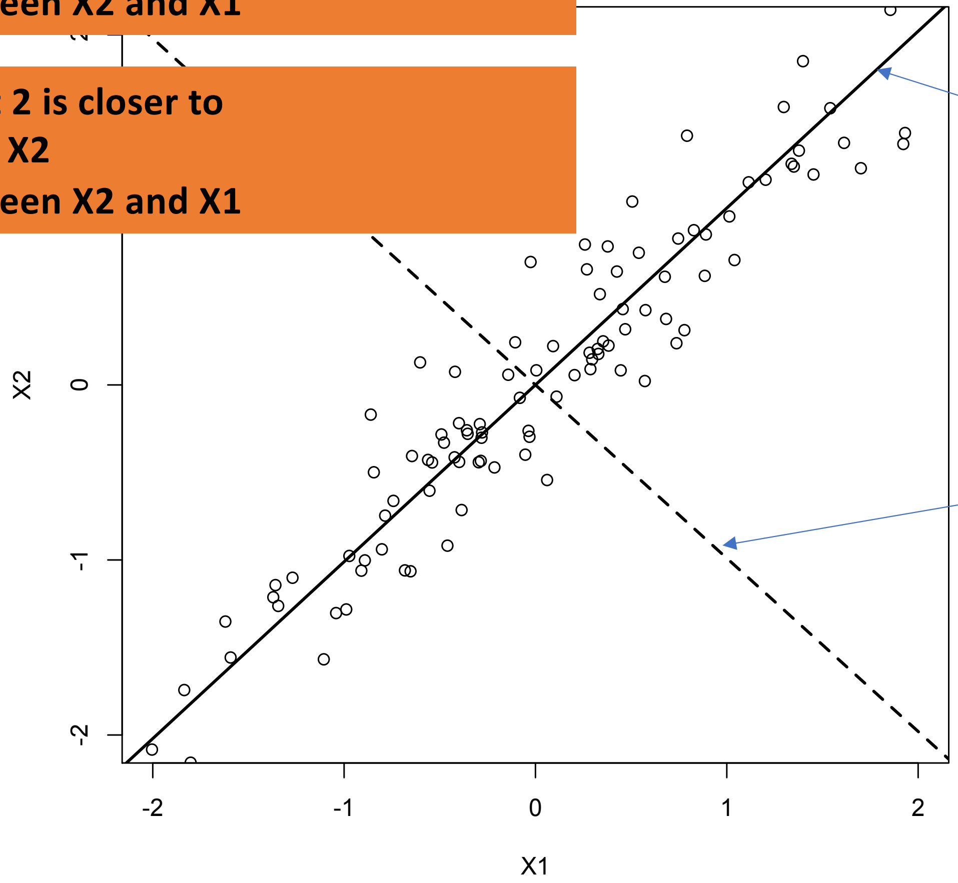
$$Z_2 = 0.711X_1 - 0.704X_2$$

Question 1: Component 1 is closer to

A. **The mean of X1 and X2**
B. **The difference between X2 and X1**

Question 2: Component 2 is closer to

A. **The mean of X1 and X2**
B. **The difference between X2 and X1**



Component 1

$$Z_1 = 0.704X_1 + 0.711X_2$$

Component 2

$$Z_2 = 0.711X_1 - 0.704X_2$$

Replace

$$Y = \alpha_1 X_1 + \alpha_2 X_2 + \varepsilon$$

by

$$\begin{aligned} Y &= b_1 CP_1 + \varepsilon \\ &= b_1(0.704X_1 + 0.711X_2) + \varepsilon \\ &= 0.097X_1 + 0.098X_2 + \varepsilon \end{aligned}$$

Example: maize biomass prediction

Objective :

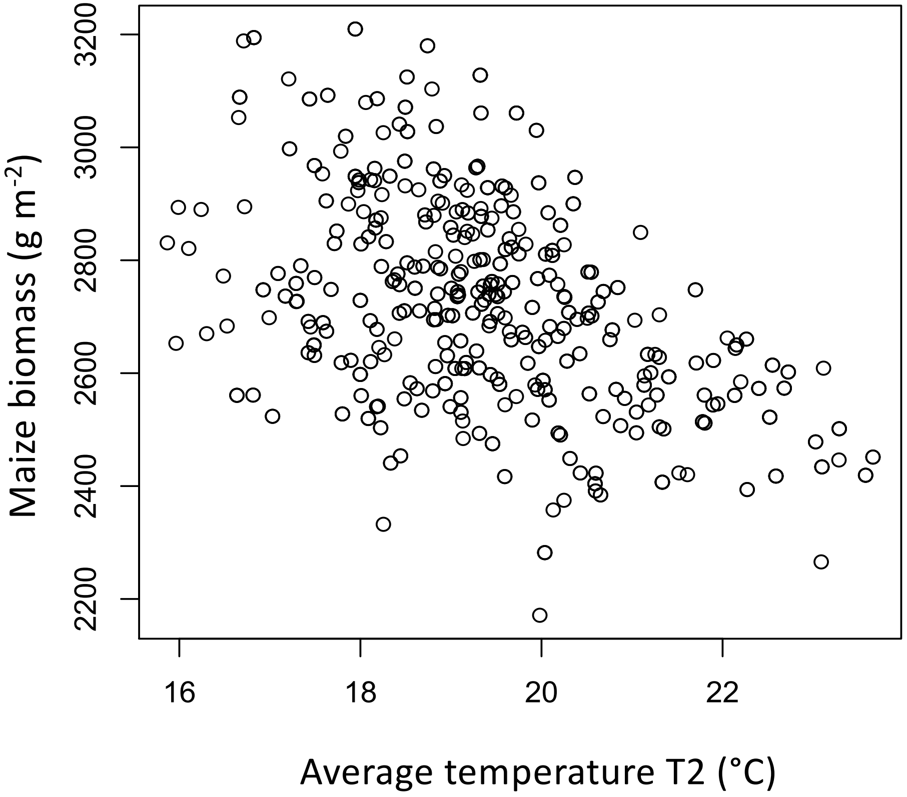
Develop a simple model predicting the final biomass of maize, noted B (g m^{-2}), from 6 input variables describing the climatic conditions during the growing season, under optimal water conditions.

Data :

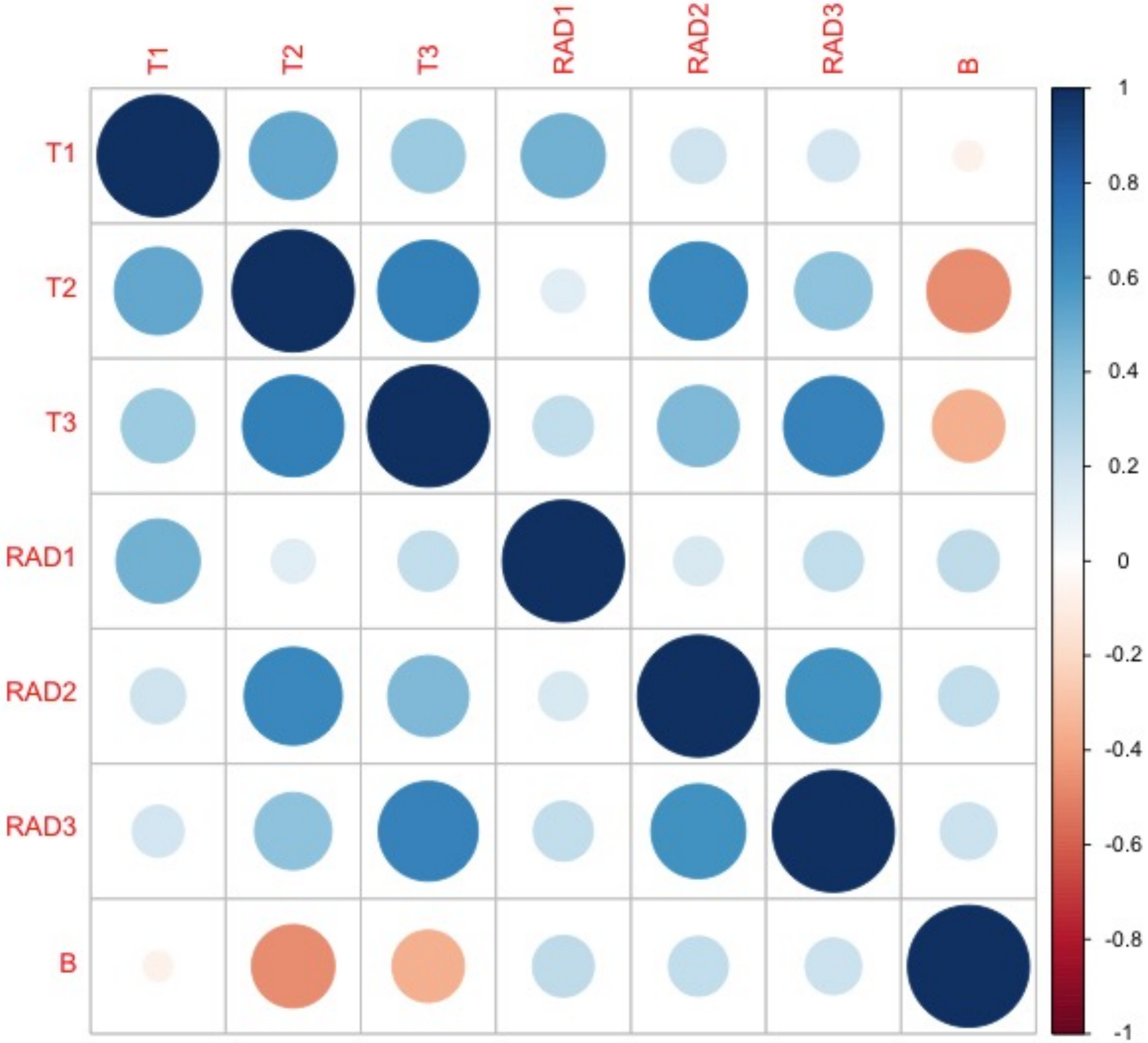
- 680 biomass data obtained for 40 different sites in France and for 17 years (1995 to 2011).
- Mean temperatures during the first part of the growing season (day 1 to day 50), during the second part of the growing season (day 51 to day 100) and during the last part of the growing season (day 101 to day 150). They are noted (T1, T2, T3).
- Average radiations during the same periods (RAD1, RAD2, RAD3).

https://github.com/davemakowski/TP_machinelearning

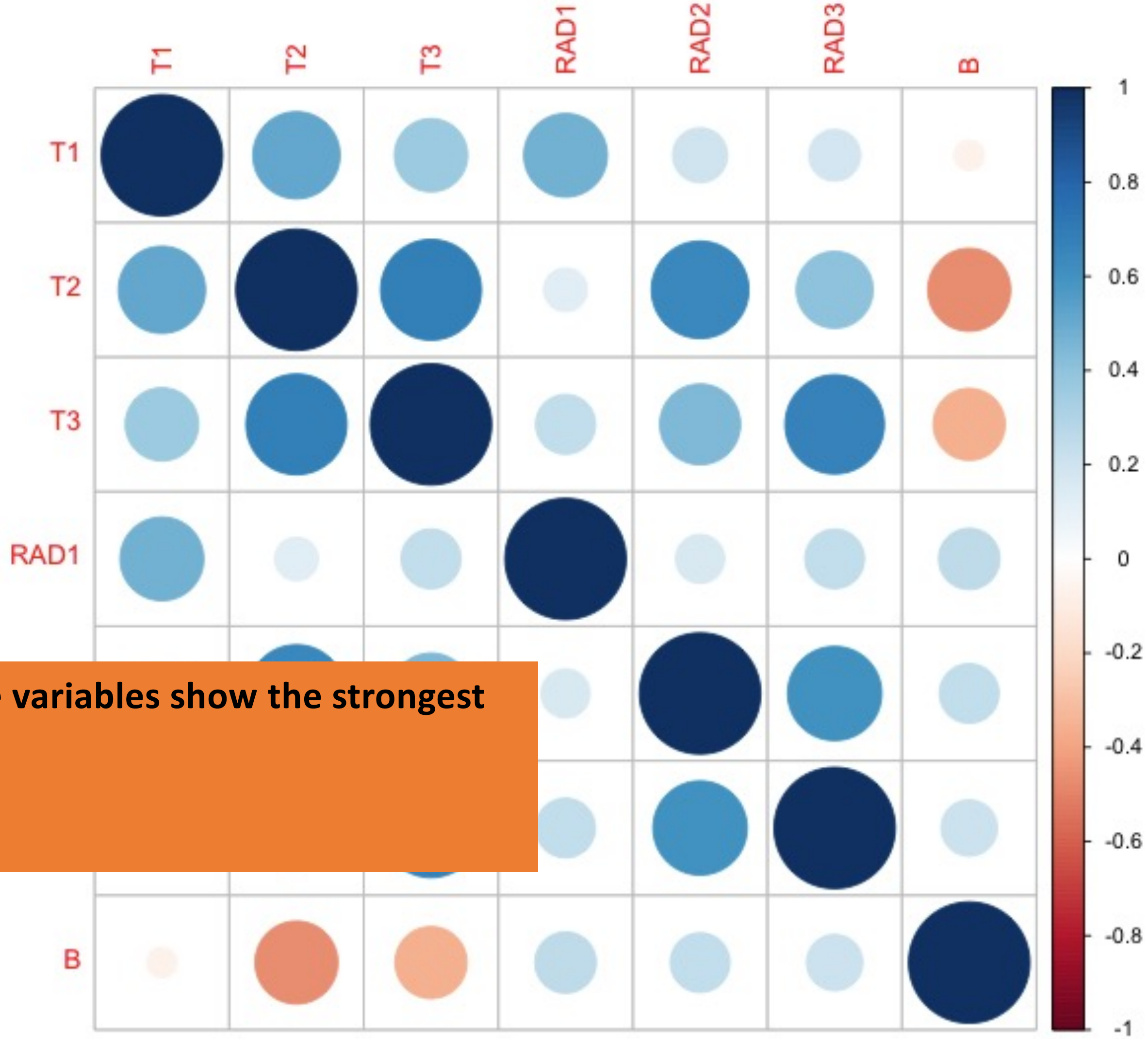
Example: maize biomass



Example: maize biomass



Example: maize biomass

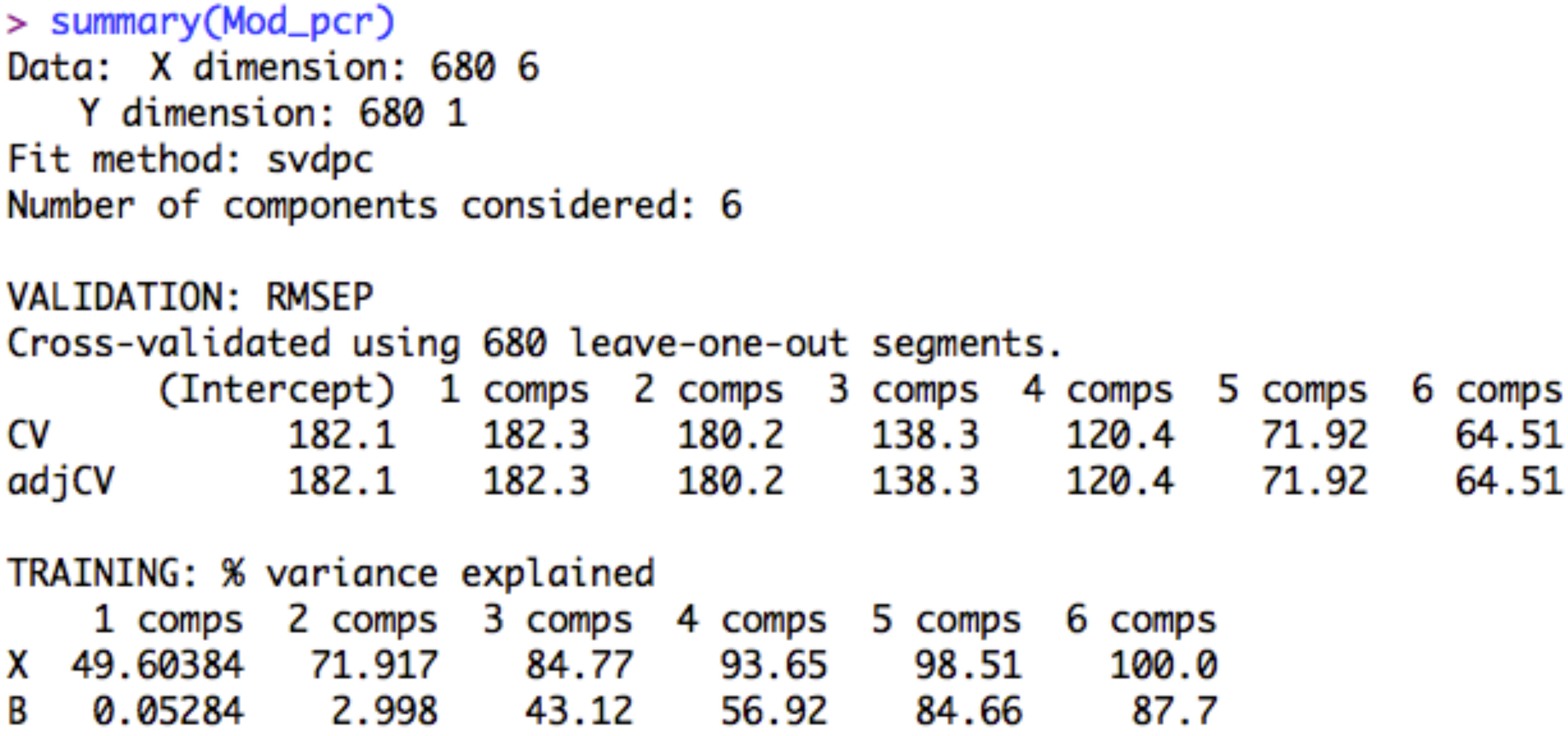


Question : Which climate variables show the strongest effects on the biomass?

A. RAD2, RAD3

B. T2, T3

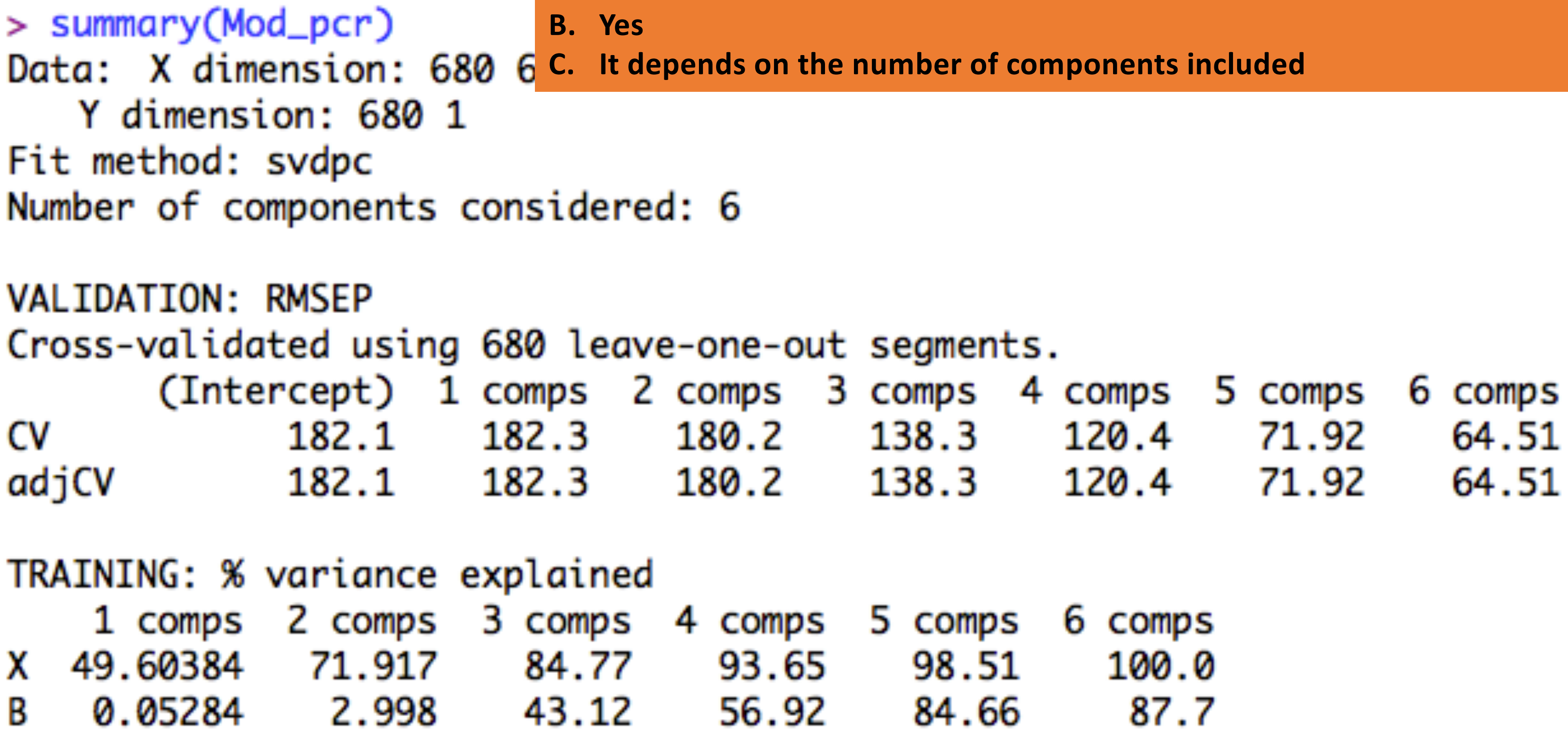
```
library(pls)

Mod_pcr<-pcr(B~T1+T2+T3+RAD1+RAD2+RAD3, data=DataSet,
validation="LOO", scale="TRUE")

summary(Mod_pcr)
```



```
> summary(Mod_pcr)
Data:   X dimension: 680 6
    Y dimension: 680 1
Fit method: svdpc
Number of components considered: 6

VALIDATION: RMSEP
Cross-validated using 680 leave-one-out segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           182.1    182.3    180.2    138.3    120.4    71.92    64.51
adjCV        182.1    182.3    180.2    138.3    120.4    71.92    64.51

TRAINING: % variance explained
    1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
X  49.60384   71.917    84.77    93.65    98.51    100.0
B   0.05284    2.998    43.12    56.92    84.66     87.7
```

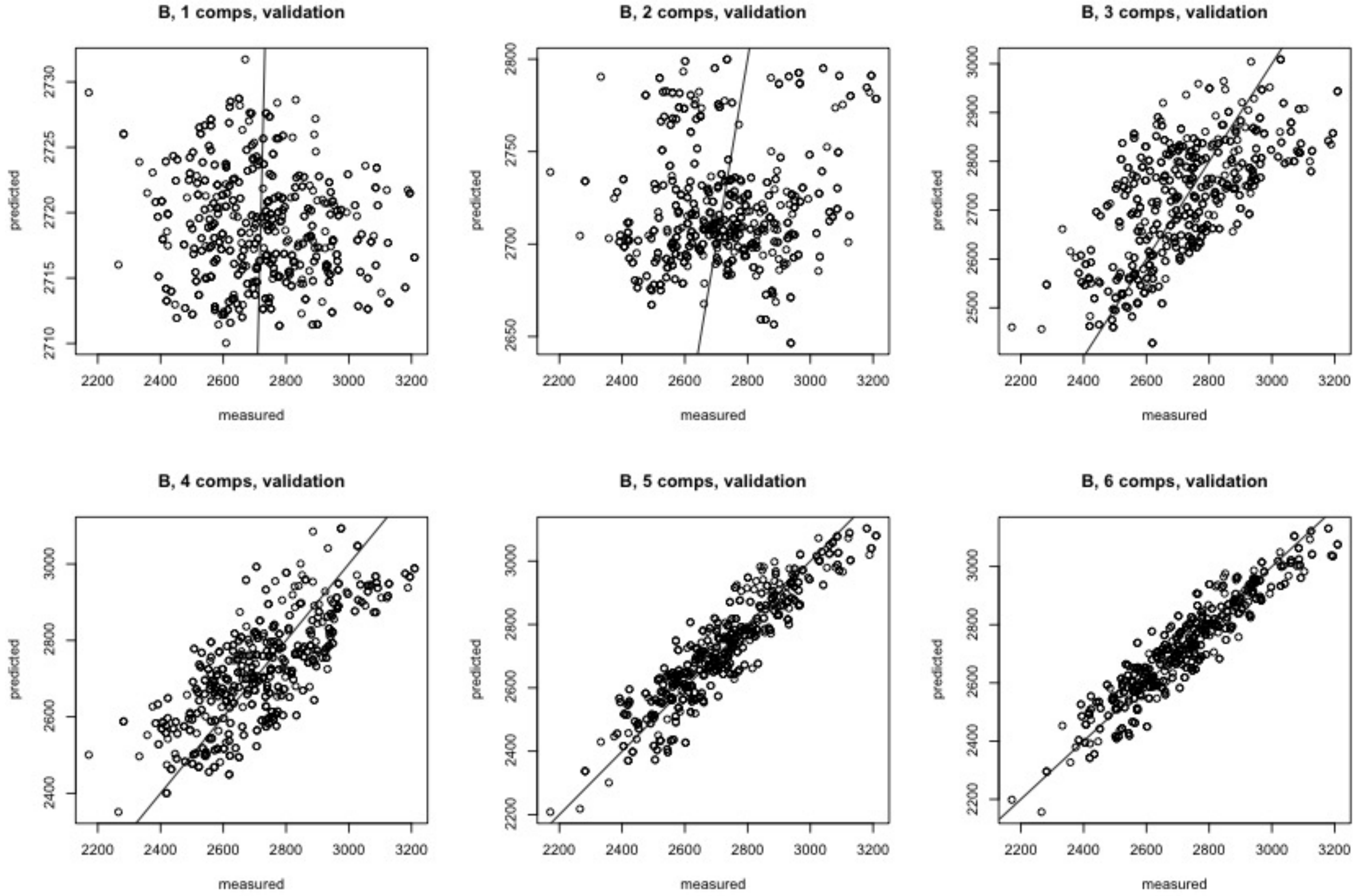
Question : Is the average error of prediction lower than 100 g/m^2 ?

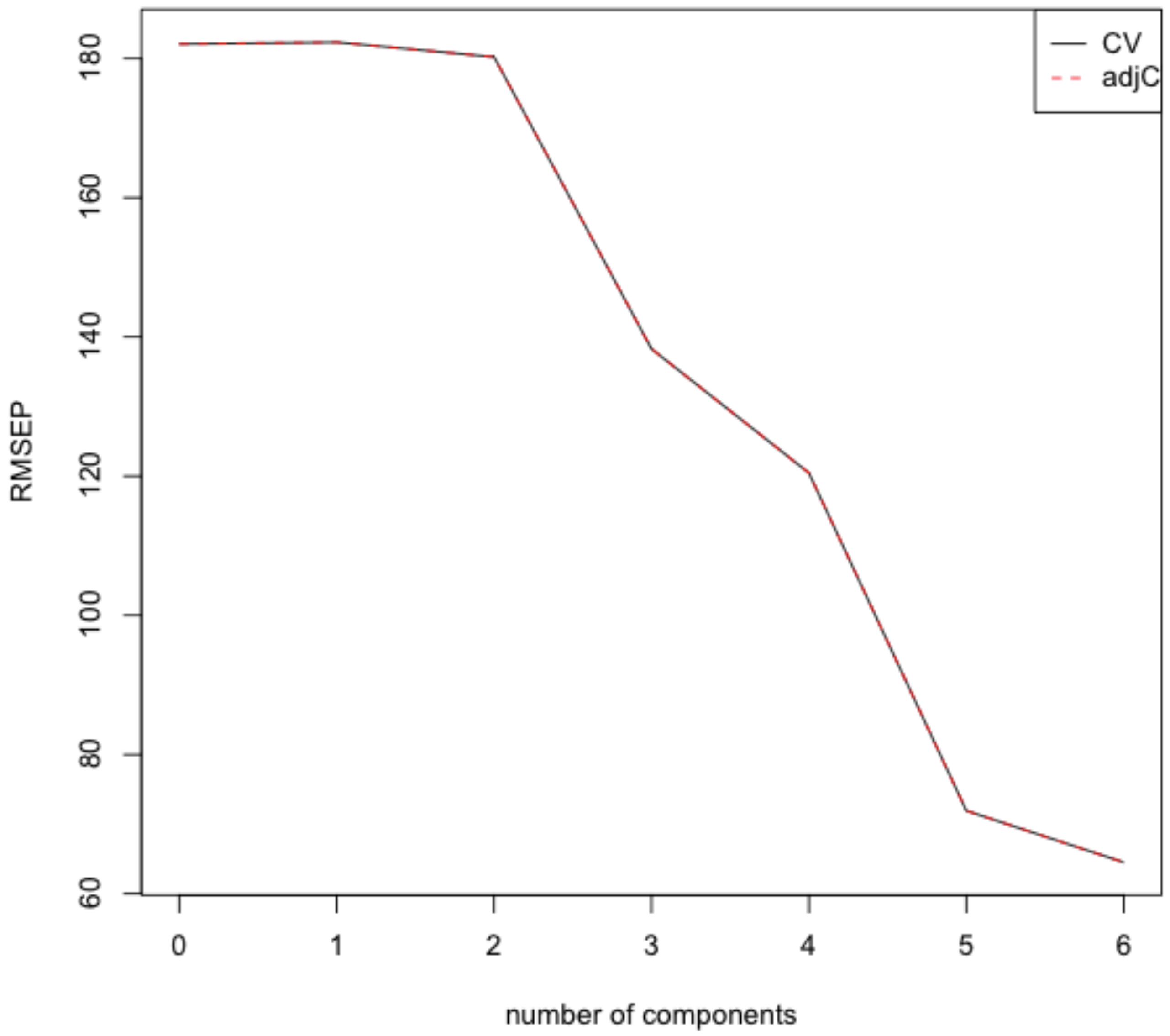
A. No
B. Yes
C. It depends on the number of components included

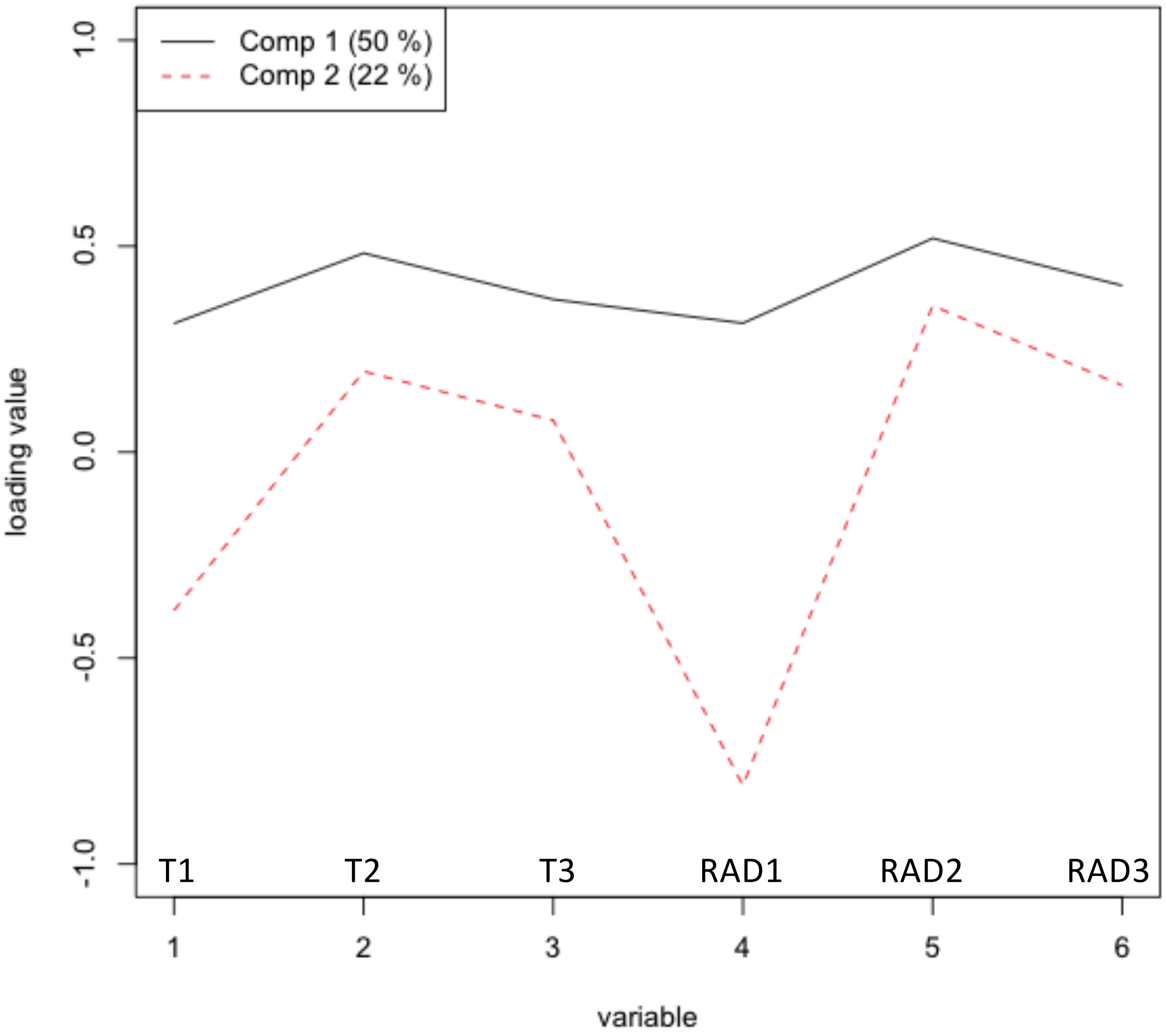


```
> summary(Mod_pcr)
Data:  X dimension: 680 6
    Y dimension: 680 1
Fit method: svdpc
Number of components considered: 6

VALIDATION: RMSEP
Cross-validated using 680 leave-one-out segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           182.1    182.3    180.2    138.3    120.4    71.92    64.51
adjCV        182.1    182.3    180.2    138.3    120.4    71.92    64.51

TRAINING: % variance explained
    1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
X  49.60384   71.917    84.77    93.65    98.51    100.0
B   0.05284    2.998    43.12    56.92    84.66     87.7
```

```r
par(mfrow=c(2,3))

plot(Mod_pcr, line=TRUE, ncomp=1)
plot(Mod_pcr, line=TRUE, ncomp=2)
plot(Mod_pcr, line=TRUE, ncomp=3)
plot(Mod_pcr, line=TRUE, ncomp=4)
plot(Mod_pcr, line=TRUE, ncomp=5)
plot(Mod_pcr, line=TRUE)

par(mfrow=c(1,1))

plot(RMSEP(Mod_pcr), legend ="topright")
plot(Mod_pcr, "loadings", comps=1:2, legendpos="topleft", ylim=c(-1,1))
```


B, 1 comps, validation
predicted
measured
2710
2715
2720
2725
2730
2200
2400
2600
2800
3000
3200
B, 2 comps, validation
predicted
measured
2650
2700
2750
2800
2200
2400
2600
2800
3000
3200
B, 3 comps, validation
predicted
measured
2500
2600
2700
2800
2900
3000
2200
2400
2600
2800
3000
3200
B, 4 comps, validation
predicted
measured
2400
2600
2800
3000
2200
2400
2600
2800
3000
3200
B, 5 comps, validation
predicted
measured
2200
2400
2600
2800
3000
2200
2400
2600
2800
3000
3200
B, 6 comps, validation
predicted
measured
2200
2400
2600
2800
3000
2200
2400
2600
2800
3000
3200


CV
adjC
RMSEP
180
160
140
120
100
80
60
0
1
2
3
4
5
6
number of components


Comp 1 (50 %)
Comp 2 (22 %)
loading value
1.0
0.5
0.0
-0.5
-1.0
T1
T2
T3
RAD1
RAD2
RAD3
1
2
3
4
5
6
variable

Difference between PCR and PLSR

- In PLSR, the components are determined to be **strongly correlated** to the values of Y
- Iterative determination of the components in PLSR
- Rarely large differences between PCR and PLSR, but PLSR often require fewer components

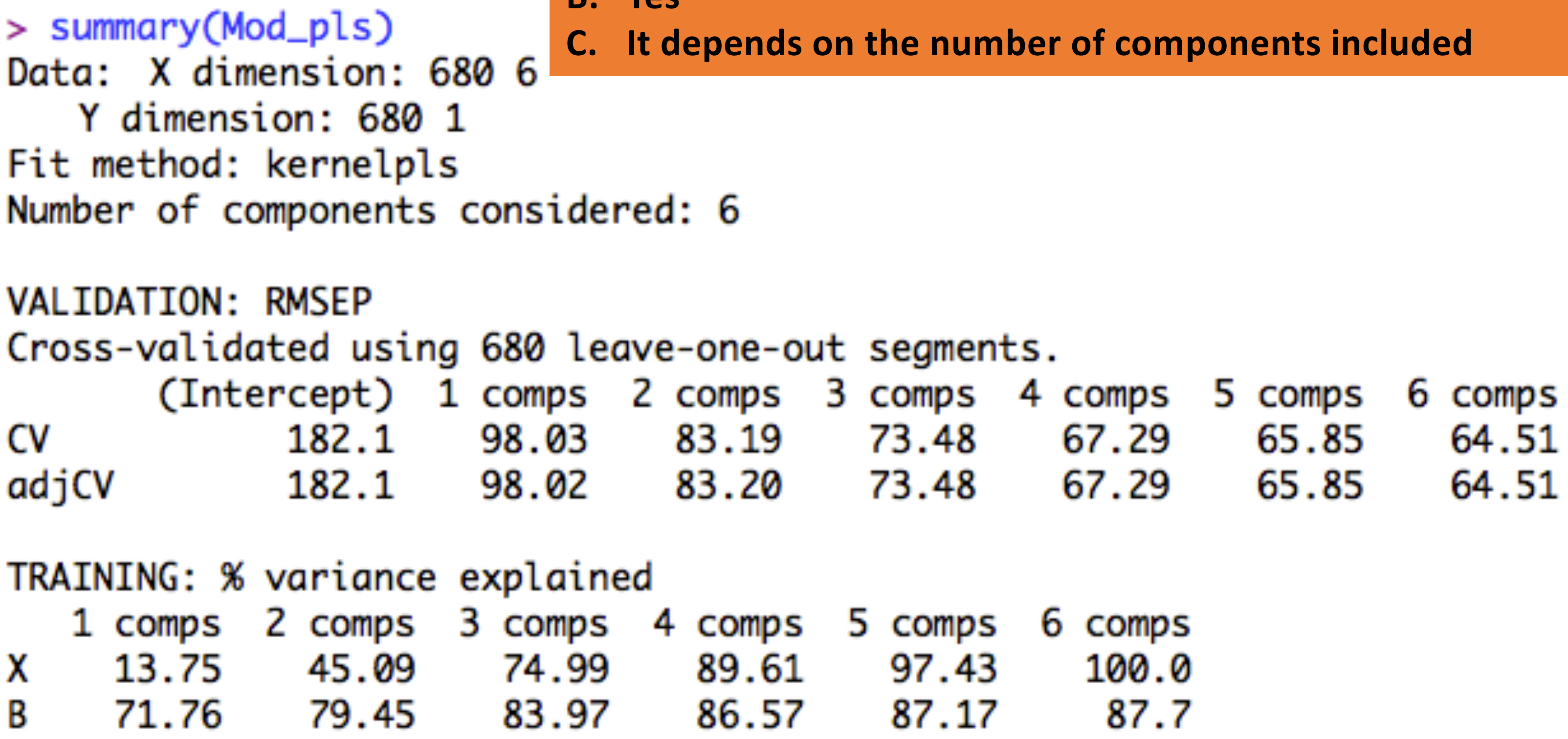
```
Mod_pls<-plsr(B~T1+T2+T3+RAD1+RAD2+RAD3, data=DataSet, validation="LOO",
scale="TRUE")

summary(Mod_pls)
```

Question : Is the average error of prediction lower than 100 g/m^2 ?

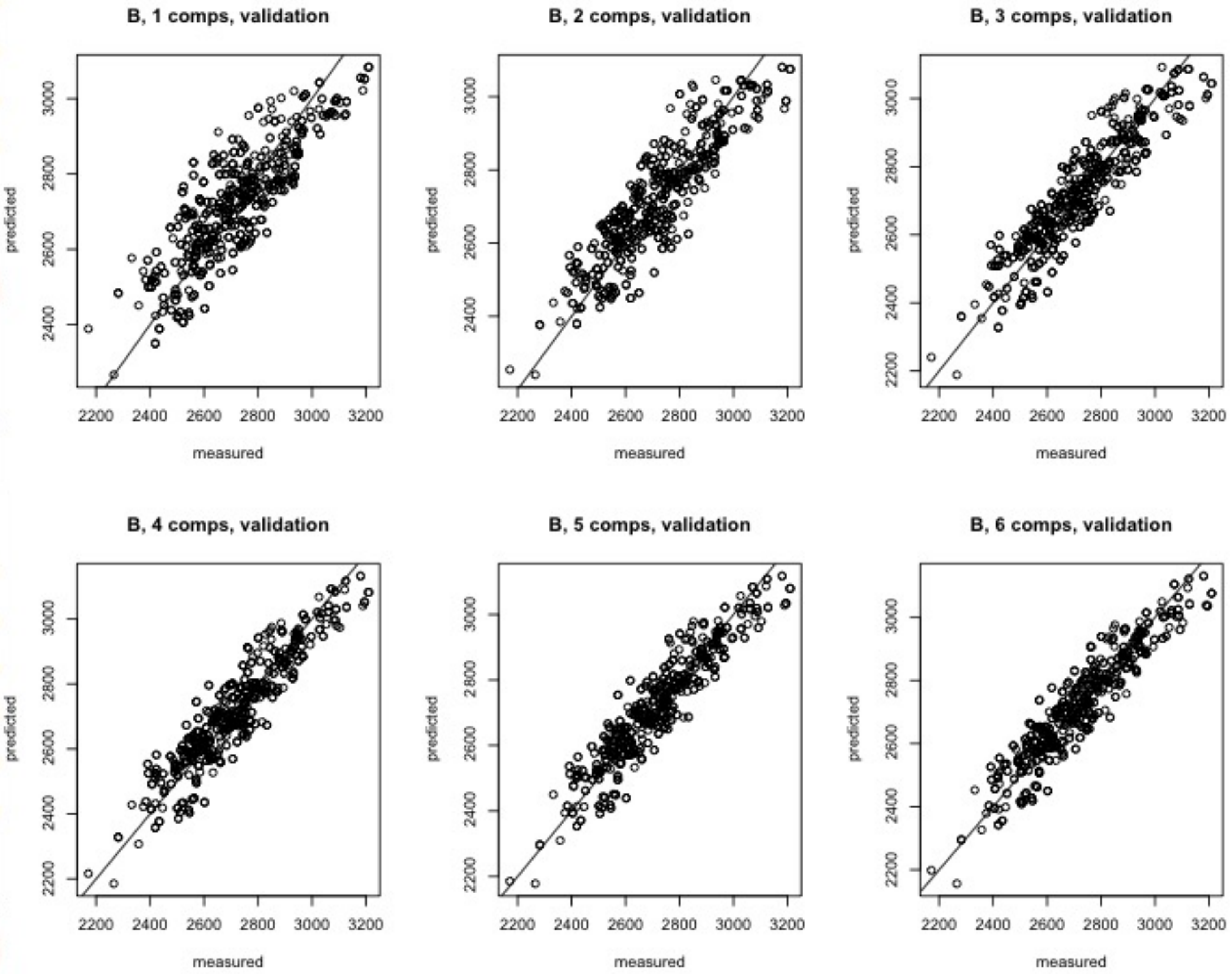
A. No

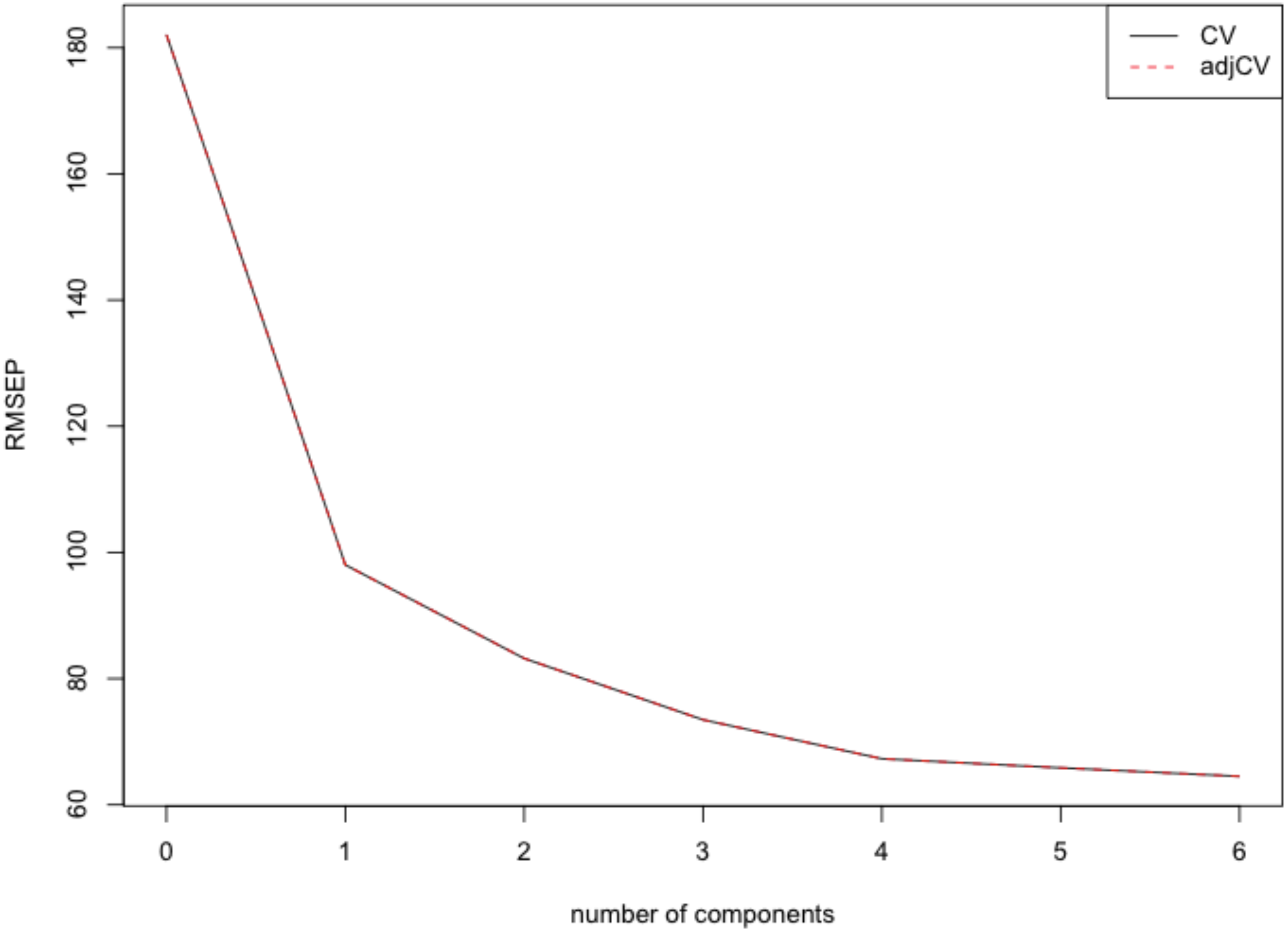
B. Yes

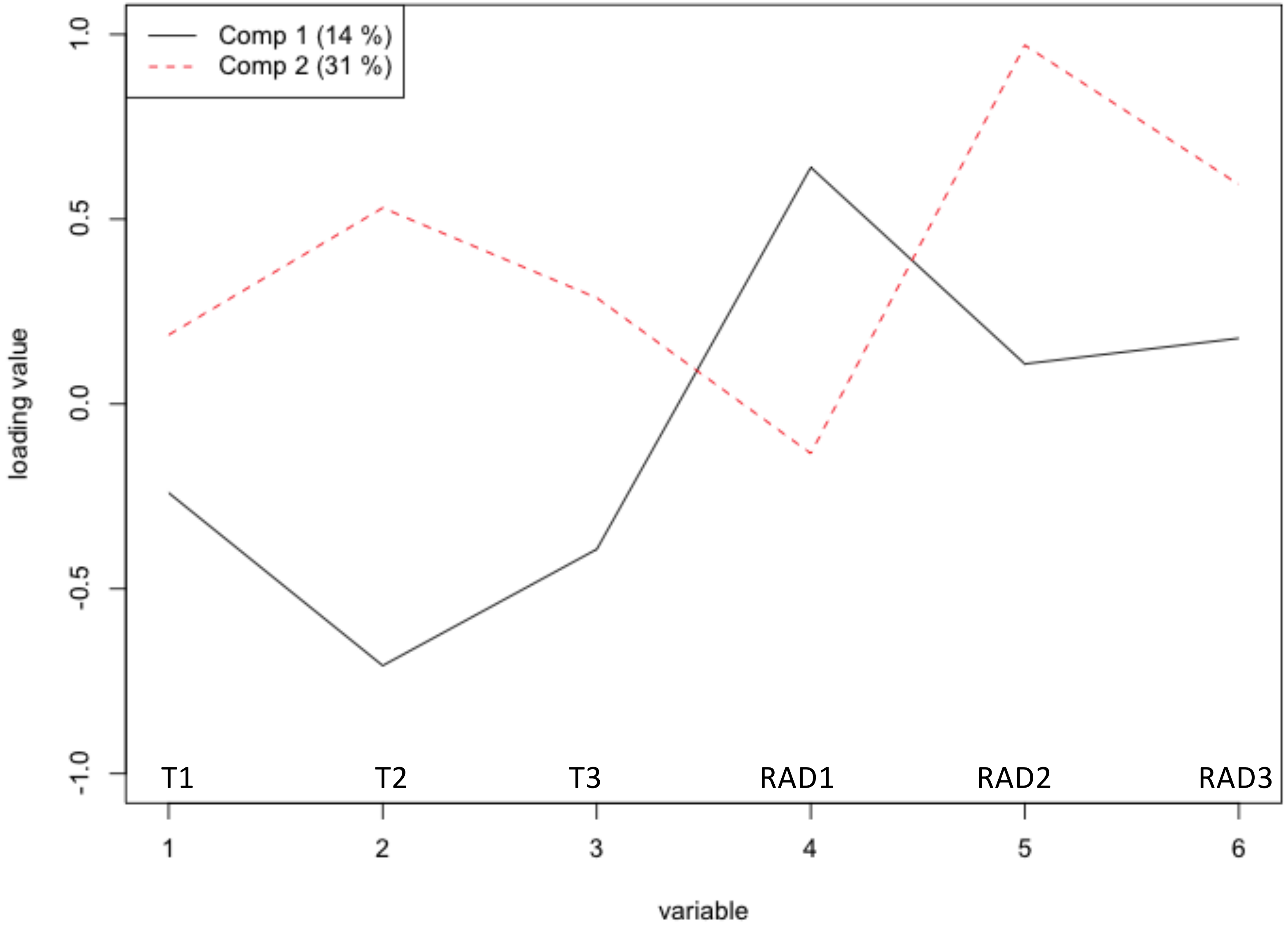
C. It depends on the number of components included

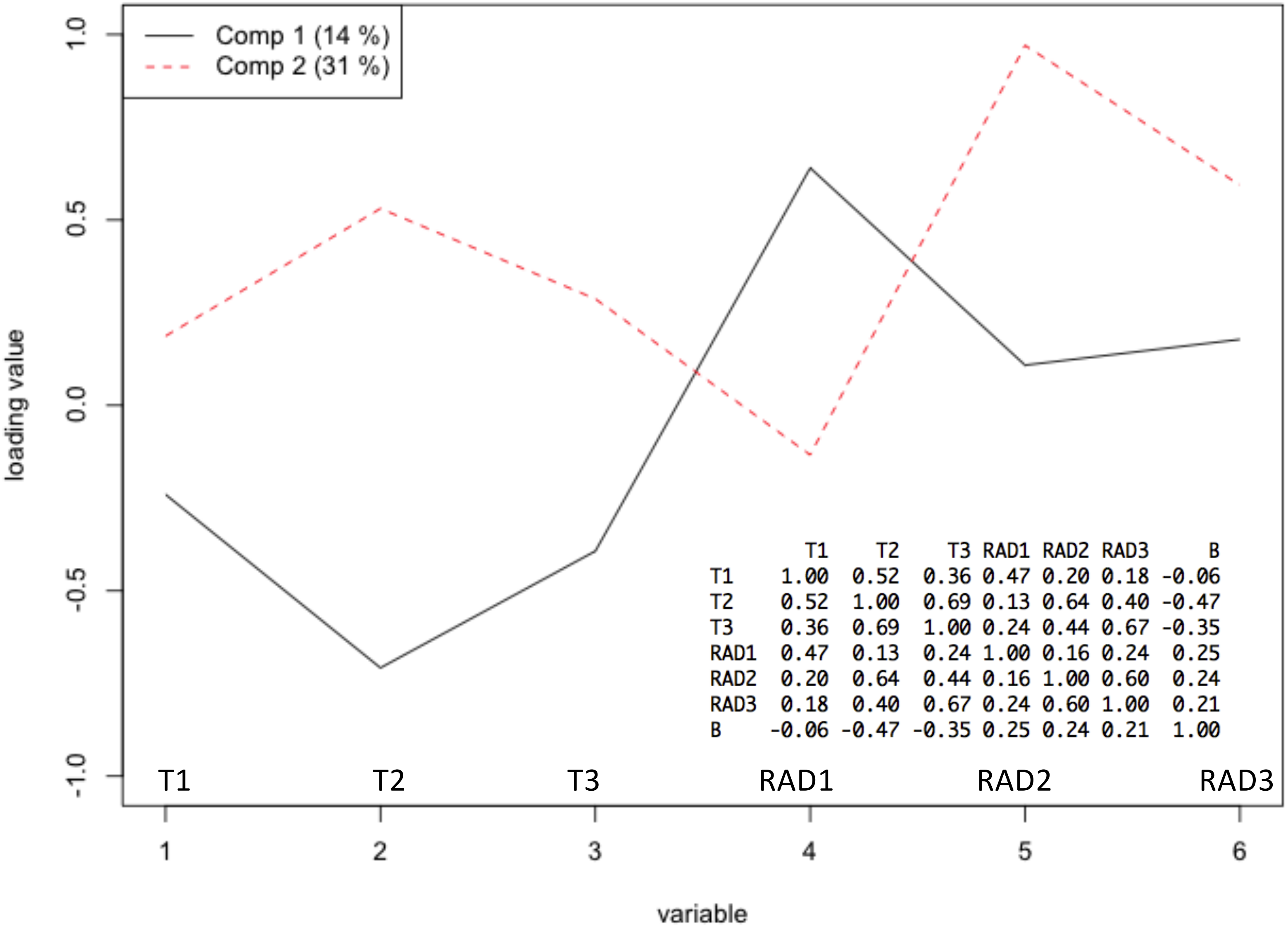


```
> summary(Mod_pls)
Data:  X dimension: 680 6
       Y dimension: 680 1
Fit method: kernelpls
Number of components considered: 6

VALIDATION: RMSEP
Cross-validated using 680 leave-one-out segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           182.1    98.03    83.19    73.48    67.29    65.85    64.51
adjCV        182.1    98.02    83.20    73.48    67.29    65.85    64.51

TRAINING: % variance explained
   1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
X    13.75    45.09    74.99    89.61    97.43    100.0
B    71.76    79.45    83.97    86.57    87.17     87.7
```


B, 1 comps, validation
B, 2 comps, validation
B, 3 comps, validation
B, 4 comps, validation
B, 5 comps, validation
B, 6 comps, validation
predicted
measured
2200
2400
2600
2800
3000
3200


CV
adjCV
RMSEP
180
160
140
120
100
80
60
0
1
2
3
4
5
6
number of components


Comp 1 (14 %)
Comp 2 (31 %)
loading value
1.0
0.5
0.0
-0.5
-1.0
T1
T2
T3
RAD1
RAD2
RAD3
1
2
3
4
5
6
variable



```
         T1    T2    T3 RAD1 RAD2 RAD3     B
T1     1.00  0.52  0.36 0.47 0.20 0.18 -0.06
T2     0.52  1.00  0.69 0.13 0.64 0.40 -0.47
T3     0.36  0.69  1.00 0.24 0.44 0.67 -0.35
RAD1   0.47  0.13  0.24 1.00 0.16 0.24  0.25
RAD2   0.20  0.64  0.44 0.16 1.00 0.60  0.24
RAD3   0.18  0.40  0.67 0.24 0.60 1.00  0.21
B     -0.06 -0.47 -0.35 0.25 0.24 0.21  1.00
```

Several issues

- Inputs X_1, …, X_P are sometimes (strongly) correlated
- **Inputs X_1, …, X_P may have non-linear effects (unknown response shape),**
- Too many inputs
- Need to estimate extreme responses, not mean response

Generalized Additive Model (GAM)

$$E(Y) = \mu + s(X_1) + \cdots + s(X_P)$$

The expected value of *Y* expressed as a smooth function of the inputs *s*(X).

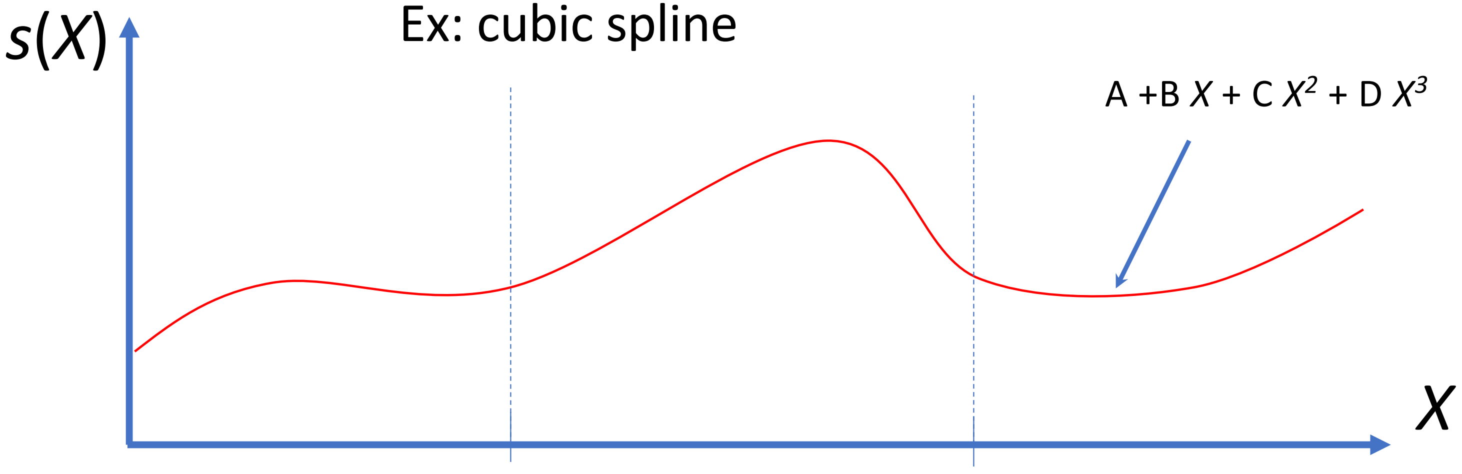
- Different types of functions *s*(X) can be used,
- Advantage of GAM : results easily interpreted.

Generalized Additive Model (GAM)

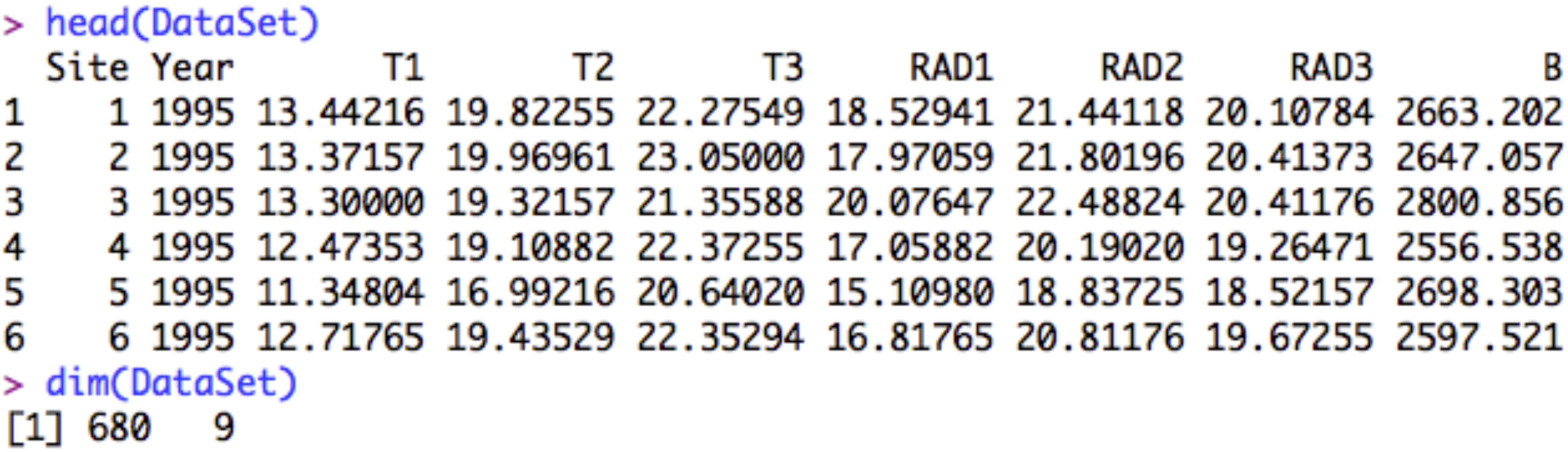
$$E(Y) = \mu + s(X_1) + \cdots + s(X_P)$$

$s(X)$ = spline = piece-wise polynomial function

Ex: cubic spline



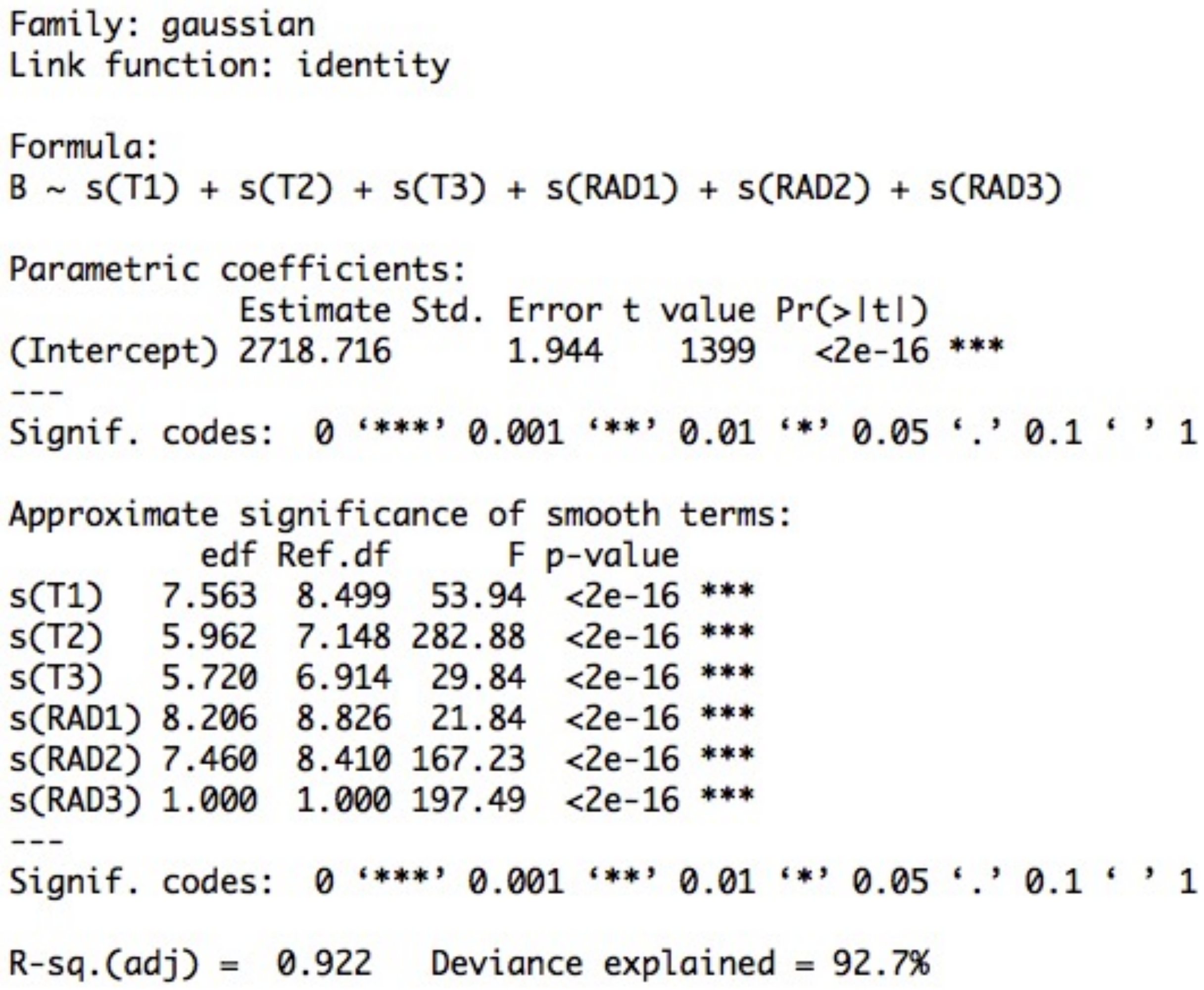
Example: maize biomass



```
> head(DataSet)
  Site Year       T1       T2       T3     RAD1     RAD2     RAD3        B
1    1 1995 13.44216 19.82255 22.27549 18.52941 21.44118 20.10784 2663.202
2    2 1995 13.37157 19.96961 23.05000 17.97059 21.80196 20.41373 2647.057
3    3 1995 13.30000 19.32157 21.35588 20.07647 22.48824 20.41176 2800.856
4    4 1995 12.47353 19.10882 22.37255 17.05882 20.19020 19.26471 2556.538
5    5 1995 11.34804 16.99216 20.64020 15.10980 18.83725 18.52157 2698.303
6    6 1995 12.71765 19.43529 22.35294 16.81765 20.81176 19.67255 2597.521
> dim(DataSet)
[1] 680   9
```

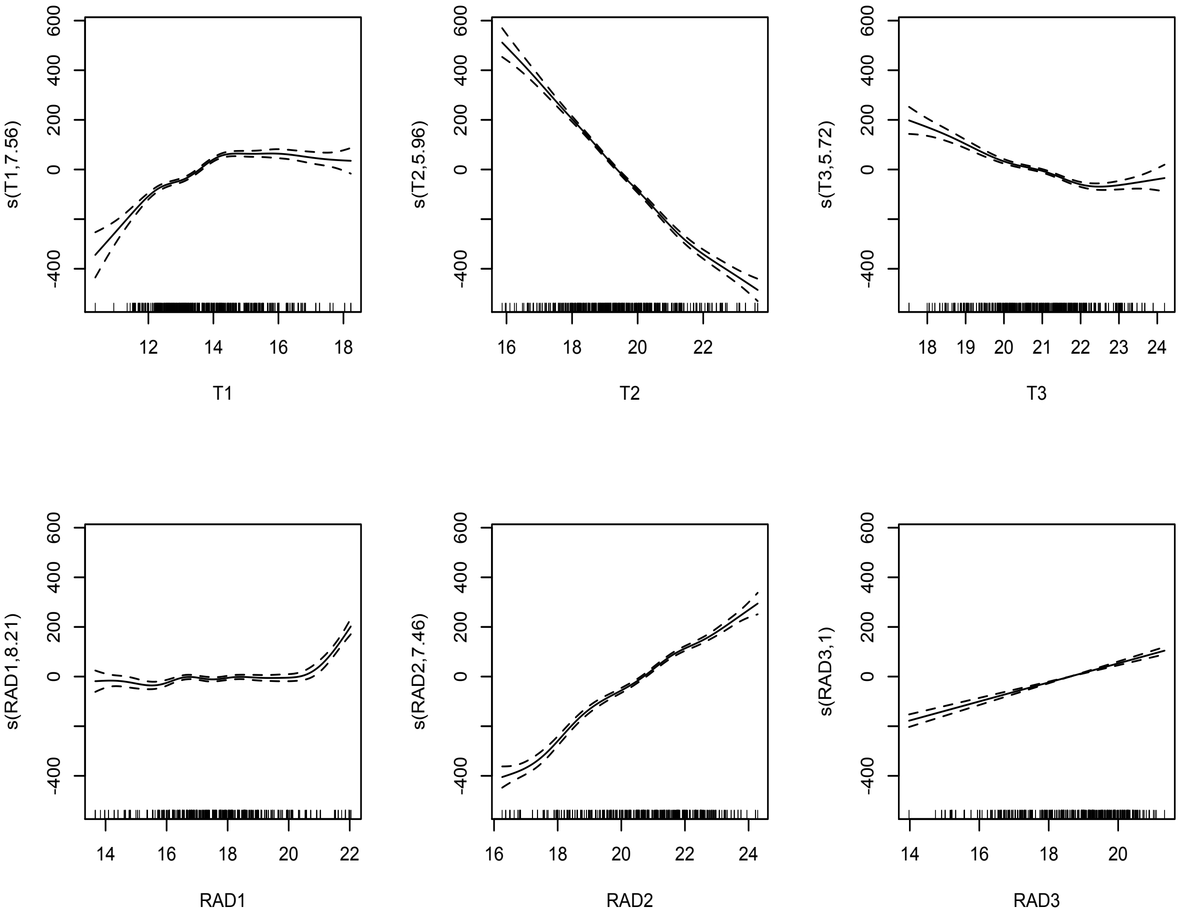
Example: maize biomass

```
library(mgcv)

Mod_gam<-
gam(B~s(T1)+s(T2)+s(T3)+s(RAD1)+s(RAD2)+s(RAD3),
data=DataSet)

summary(Mod_gam)
plot(Mod_gam)
```



```
Family: gaussian
Link function: identity

Formula:
B ~ s(T1) + s(T2) + s(T3) + s(RAD1) + s(RAD2) + s(RAD3)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 2718.716      1.944    1399   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
          edf Ref.df      F p-value
s(T1)   7.563  8.499  53.94  <2e-16 ***
s(T2)   5.962  7.148 282.88  <2e-16 ***
s(T3)   5.720  6.914  29.84  <2e-16 ***
s(RAD1) 8.206  8.826  21.84  <2e-16 ***
s(RAD2) 7.460  8.410 167.23  <2e-16 ***
s(RAD3) 1.000  1.000 197.49  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.922   Deviance explained = 92.7%
```

Example: maize biomass



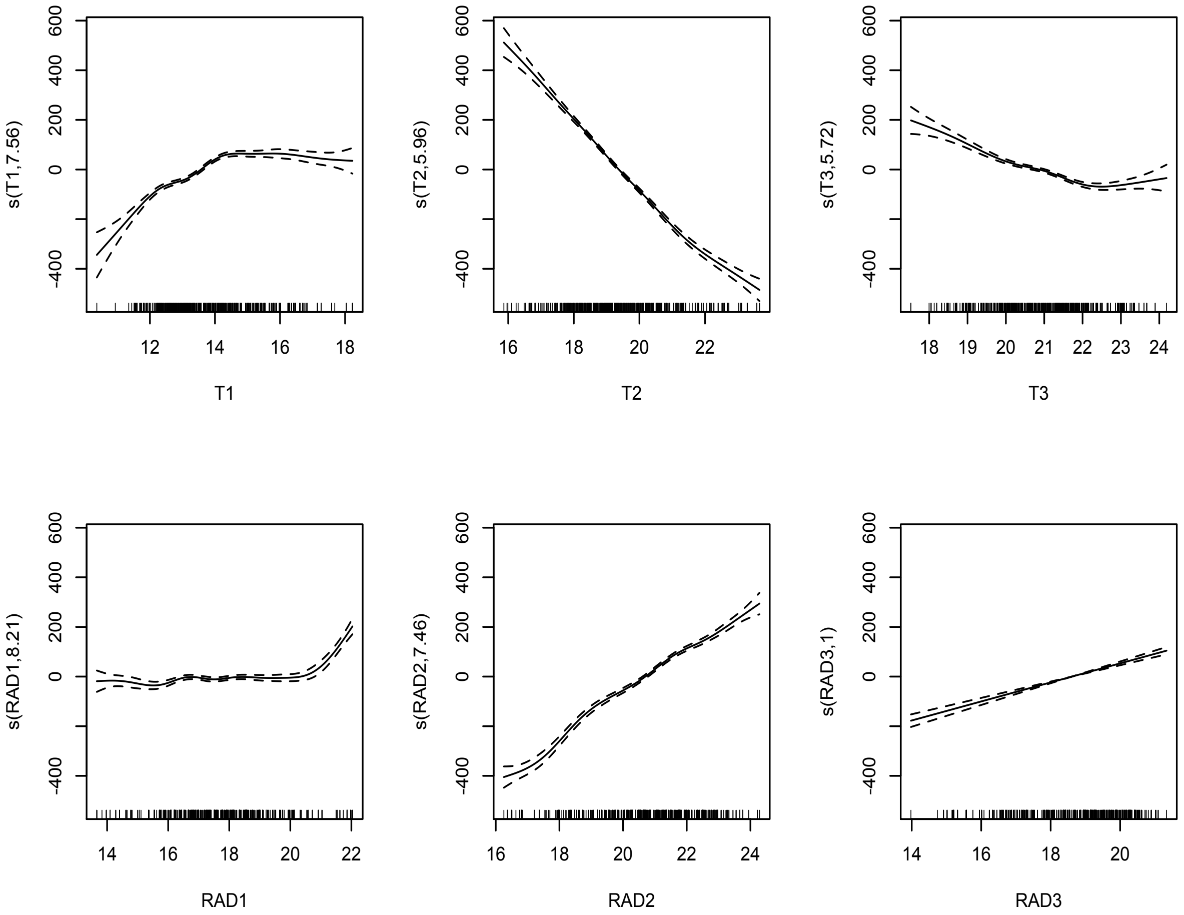
Example: maize biom

Question : Is the effect of T1 on the biomass linear ?

A. No

B. Yes

C. Almost



Example: maize biom

Question : Is the effect of T1 on the biomass linear?

A. No

B. Yes

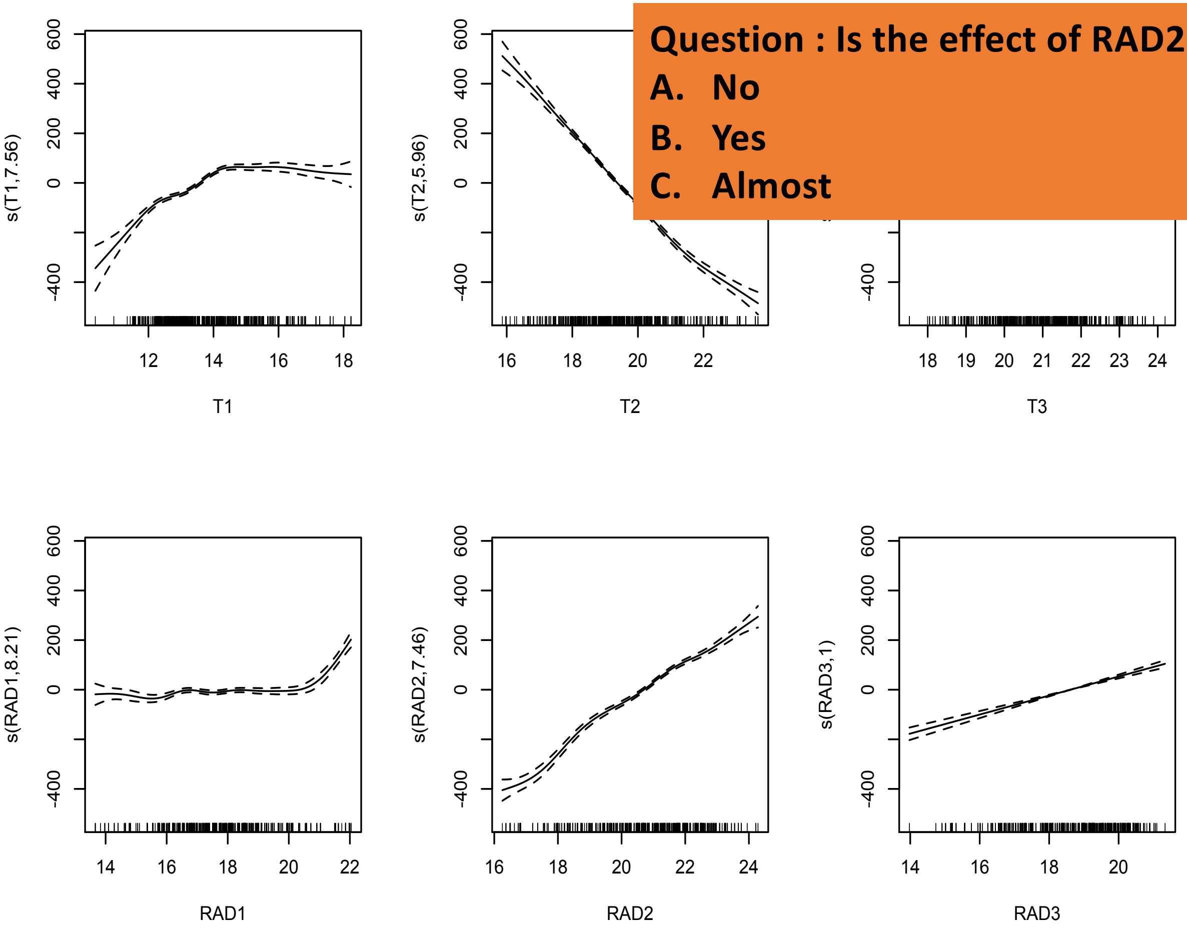
C. Almost

Question : Is the effect of RAD2 on the biomass linear?

A. No

B. Yes

C. Almost



```r
RMSE_gam<-sqrt(mean((DataSet$B-predict(Mod_gam))^2))
RMSE_gam


#Cross-validation
B_pred_gam<-rep(NA,length(DataSet$B))


List_year<-unique(DataSet$Year)


for (i in 1:length(List_year))
{
Training_i<-DataSet[DataSet$Year!=List_year[i],]
Test_i<-DataSet[DataSet$Year==List_year[i],]
Mod_i<-gam(B~s(T1)+s(T2)+s(T3)+s(RAD1)+s(RAD2)+s(RAD3), data=Training_i)
B_gam_i<-predict(Mod_i, newdata=Test_i)
B_pred_gam[DataSet$Year==List_year[i]]<-B_gam_i
}


RMSEP_gam<-sqrt(mean((DataSet$B-B_pred_gam)^2))
RMSEP_gam
```

```r
RMSE_gam<-sqrt(mean((DataSet$B-predict(Mod_gam))^2))
RMSE_gam


#Cross-validation
B_pred_gam<-rep(NA,length(DataSet$B))


List_year<-unique(DataSet$Year)
```

Question : The training dataset includes

A. All data

B. All data but the ith

C. Only the ith data

```r
for (i in 1:length(List_year))
{
Training_i<-DataSet[DataSet$Year!=List_year[i],]
Test_i<-DataSet[DataSet$Year==List_year[i],]
Mod_i<-gam(B~s(T1)+s(T2)+s(T3)+s(RAD1)+s(RAD2)+s(RAD3), data=Training_i)
B_gam_i<-predict(Mod_i, newdata=Test_i)
B_pred_gam[DataSet$Year==List_year[i]]<-B_gam_i
}


RMSEP_gam<-sqrt(mean((DataSet$B-B_pred_gam)^2))
RMSEP_gam
```

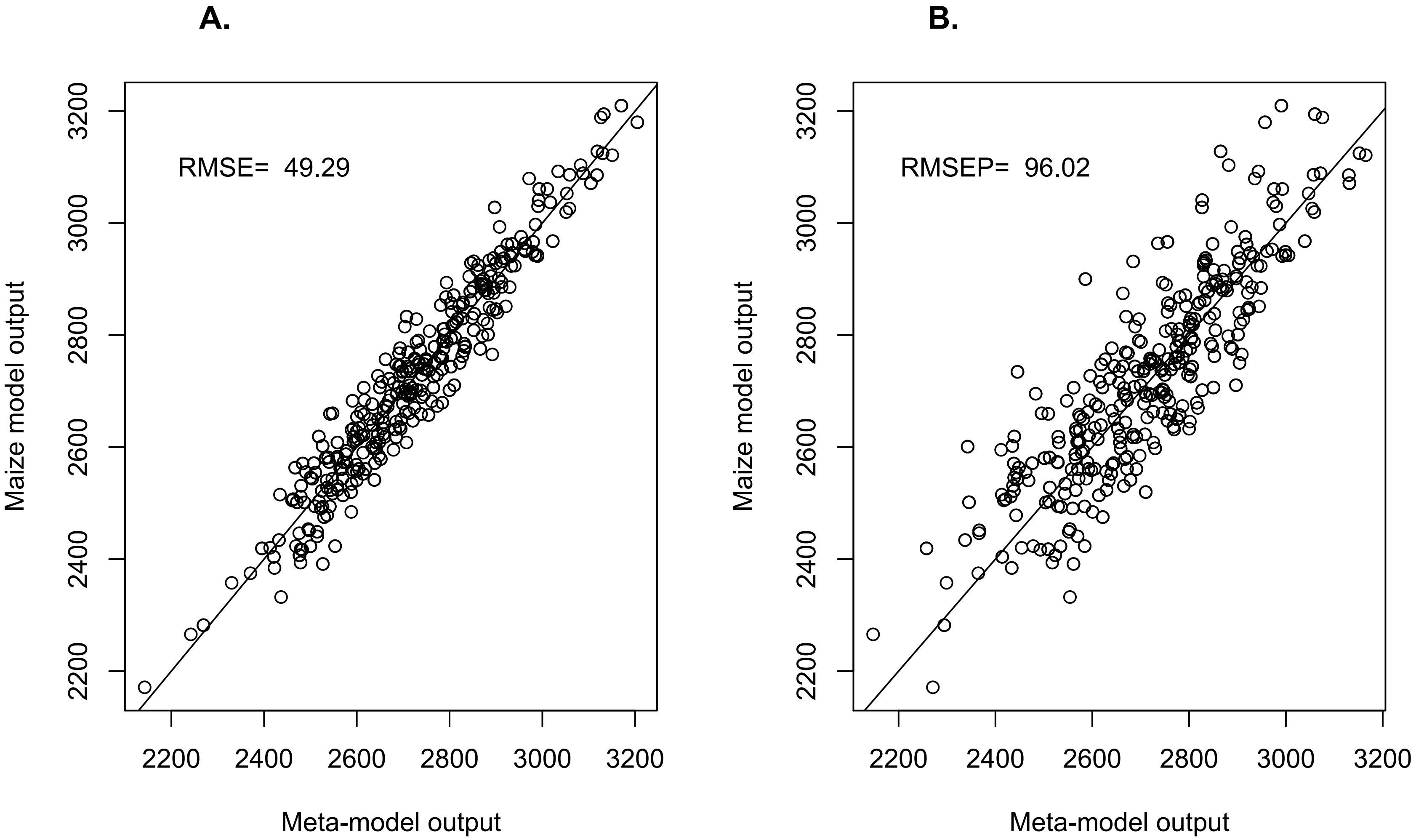
```
RMSE_gam<-sqrt(mean((DataSet$B-predict(Mod_gam))^2))
RMSE_gam


#Cross-validation
B_pred_gam<-rep(NA,length(DataSet$B))


List_year<-unique(DataSet$Year)


for (i in 1:length(List_year))
{
Training_i<-DataSet[DataSet$Year!=List_year[i],]
Test_i<-DataSet[DataSet$Year==List_year[i],]
Mod_i<-gam(B~s(T1)+s(T2)+s(T3)+s(RAD1)+s(RAD2)+s(RAD3), data=Training_i)
B_gam_i<-predict(Mod_i, newdata=Test_i)
B_pred_gam[DataSet$Year==List_year[i]]<-B_gam_i
}


RMSEP_gam<-sqrt(mean((DataSet$B-B_pred_gam)^2))
RMSEP_gam
```

Question : The test dataset includes

A. All data

B. All data but the i^{th}

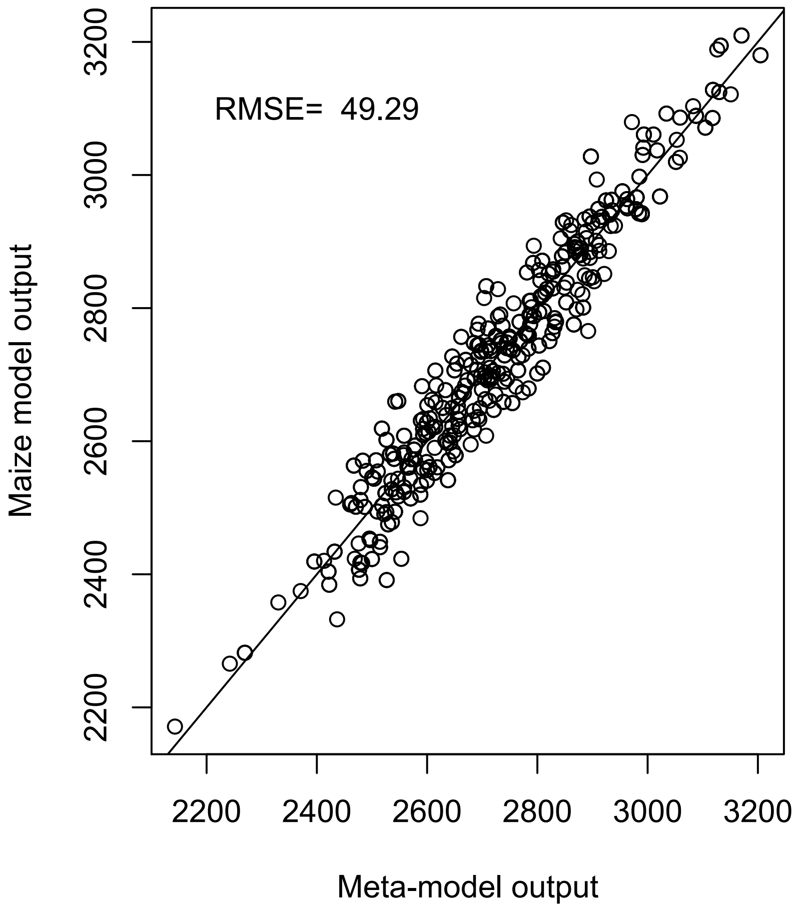
C. Only the i^{th} data

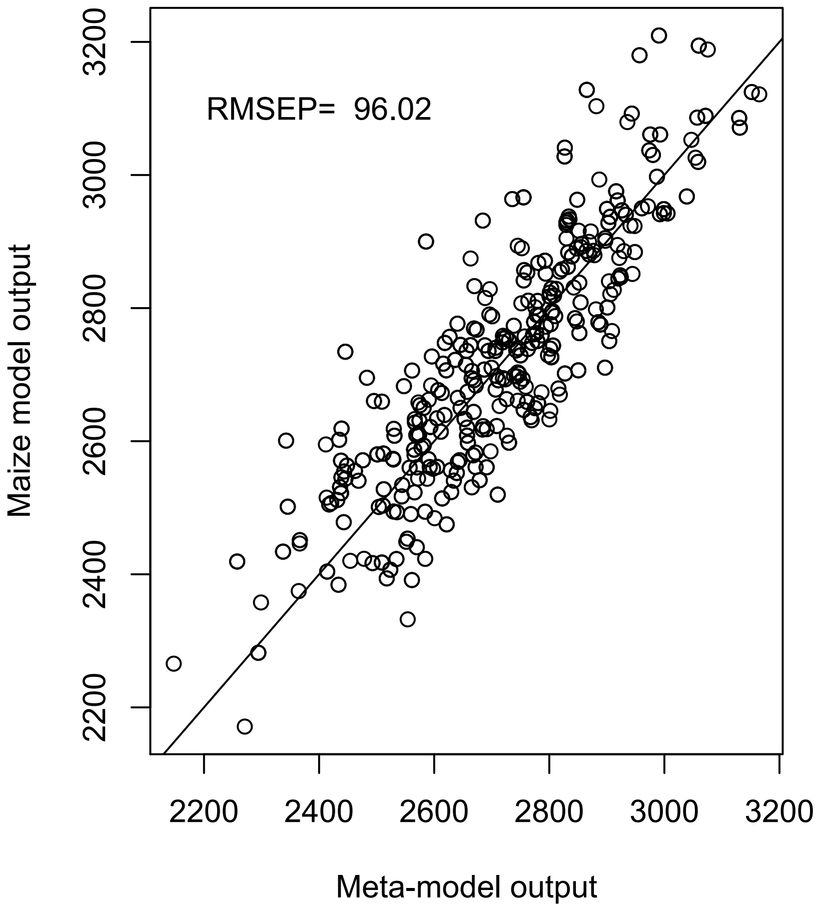

A.
RMSE= 49.29
Maize model output
Meta-model output
2200
2400
2600
2800
3000
3200
B.
RMSEP= 96.02
Maize model output
Meta-model output
2200
2400
2600
2800
3000
3200

Question : Are the prediction errors lower than 50 g/m^2 in average?

A. Yes

B. No





Several issues

- Inputs X_1, …, X_P are sometimes (strongly) correlated
- Inputs X_1, …, X_P may have non-linear effects (unknown response shape),
- **Too many inputs**
- Need to estimate extreme responses, not mean response

Other powerful regression techniques: Penalized regression methods

LASSO, ridge and elastic net

$$Y = X\theta + \varepsilon$$

$$\hat{Y} = X\hat{\theta}$$

$\hat{\theta}$ est estimé en minimisant $\left. \sum_i \left(Y_i - \hat{Y}_i\right)^2 + \lambda G \right\}$ Penalty term

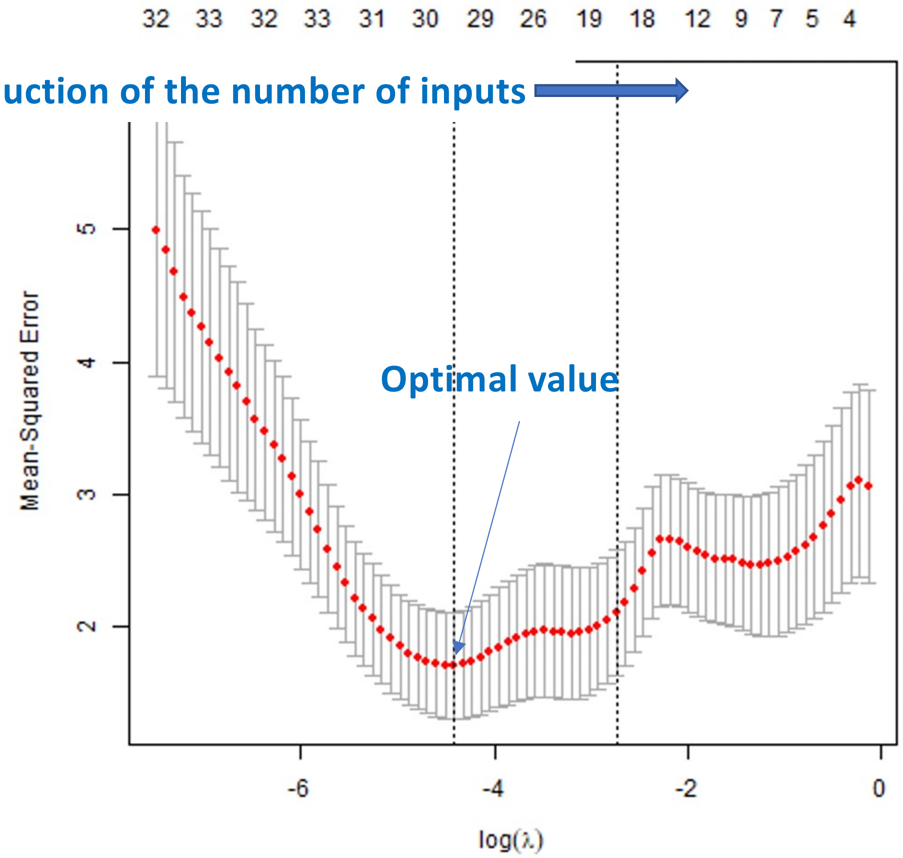
Additional parameter
(hyper-parameter)

$G = \sum_j |\theta_j|$ LASSO

$G = \sum_j \theta_j^2$ Ridge

$G = \sum_j \alpha|\theta_j| + \sum_j (1-\alpha)\theta_j^2$ Elastic net

Penalty level (hyper-parameter) optimized by cross validation



Reduction of the number of inputs

Optimal value

Example « spores »

Y=quantity of fungus spores
X=34 inputs representing climatic conditions

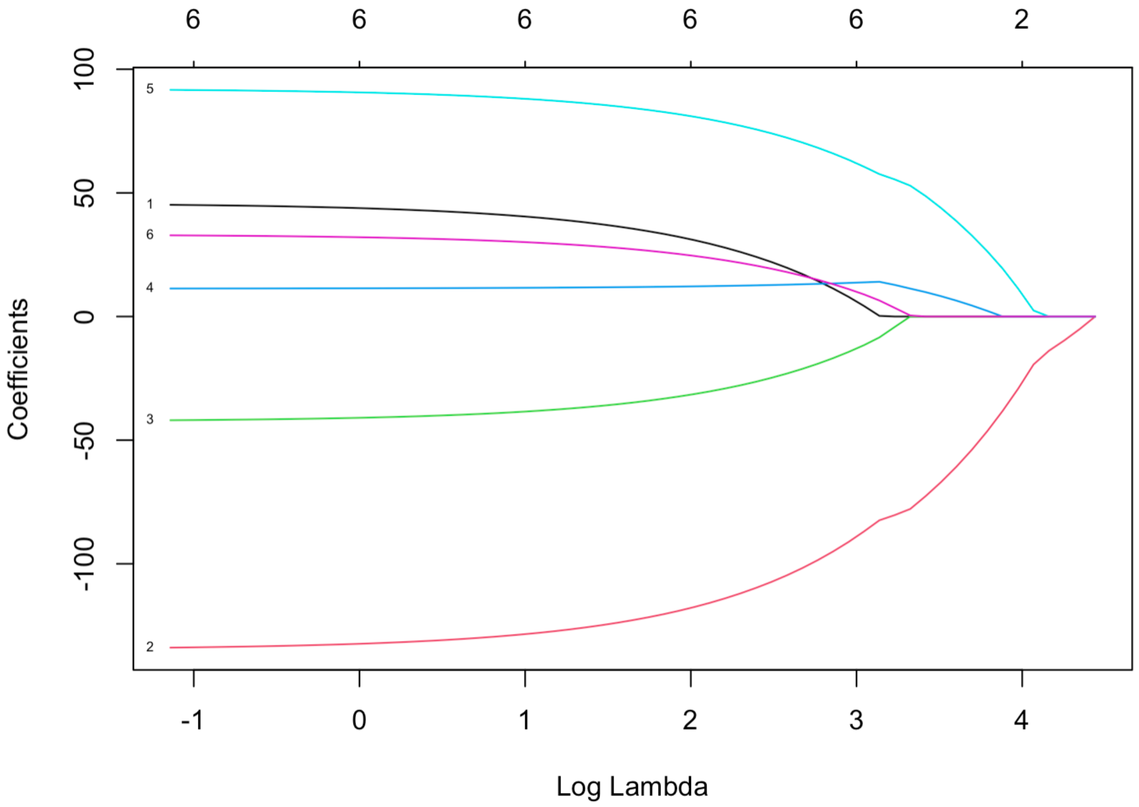
From Data science pour l'agriculture et l'environnement - Méthodes et applications avec R et Python. Editions Ellipses. https://www.editions-ellipses.fr/

R package glmnet

```
cv <- cv.glmnet(x, y, alpha = 1)
model <- glmnet(x, y, alpha = 1, lambda = cv$lambda.min)
```

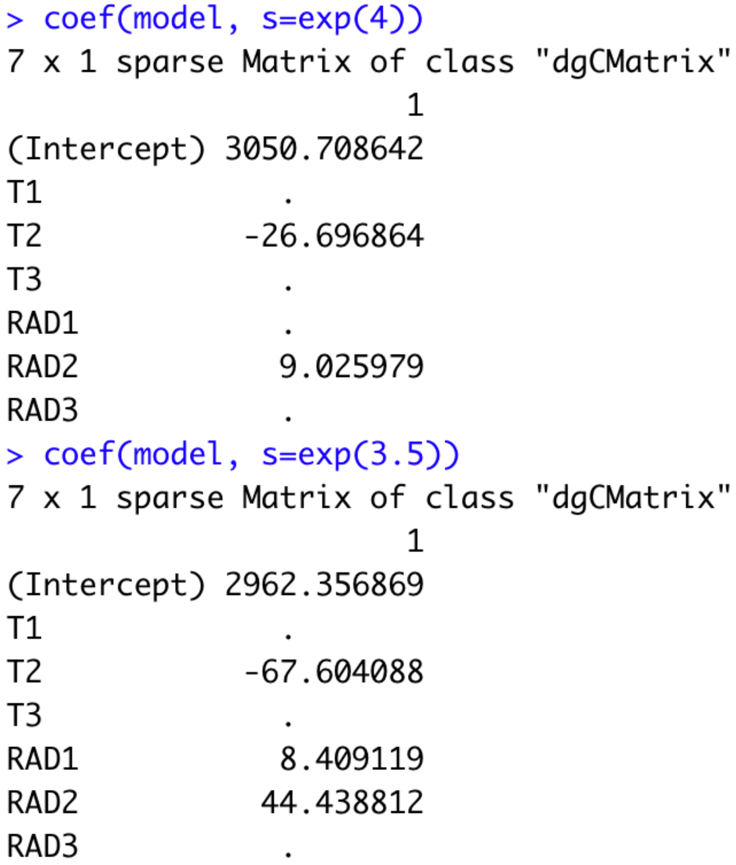
Example: maize biomass

```
library(glmnet)
X=as.matrix(DataSet[,3:8])
Y=DataSet$B

#Fit LASSO
model <- glmnet(X, Y, alpha = 1)

#Plot coefficients for different penalty levels
plot(model, xvar="lambda", label=TRUE)
```



#Regression coefficients for two levels of penalty

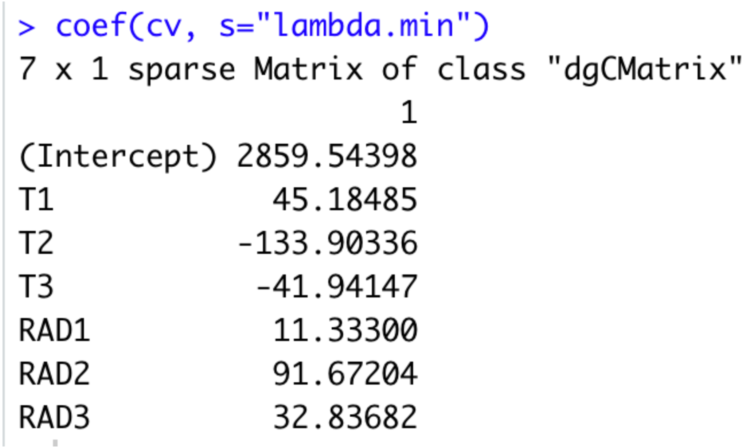
```
coef(model, s=exp(4))
coef(model, s=exp(3.5))
```



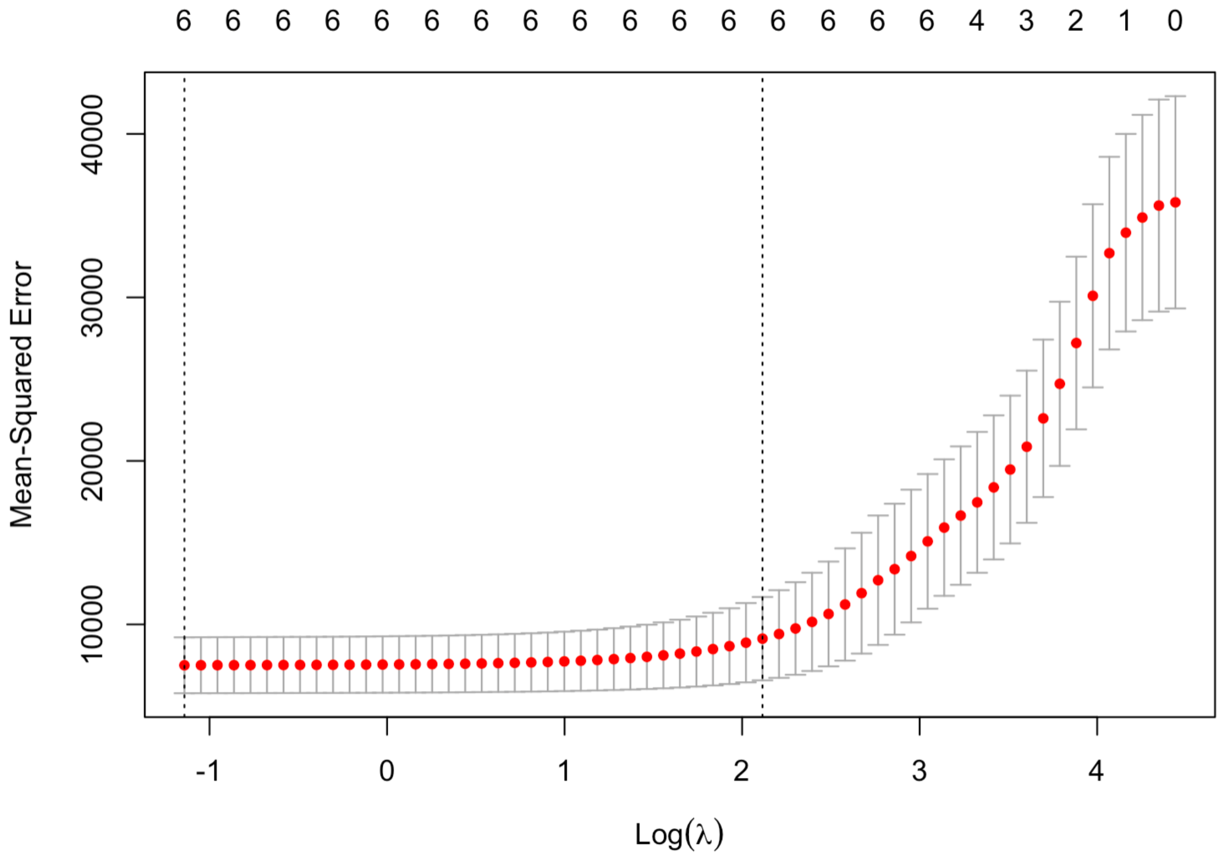
```
> coef(model, s=exp(4))
7 x 1 sparse Matrix of class "dgCMatrix"
                      1
(Intercept) 3050.708642
T1                .
T2           -26.696864
T3                .
RAD1              .
RAD2           9.025979
RAD3              .
> coef(model, s=exp(3.5))
7 x 1 sparse Matrix of class "dgCMatrix"
                      1
(Intercept) 2962.356869
T1                .
T2           -67.604088
T3                .
RAD1           8.409119
RAD2          44.438812
RAD3              .
```

#Selection of a lambda value by year-by-year cross-validation

```
cv <- cv.glmnet(X, Y, alpha=1, foldid=as.numeric(as.factor(DataSet$Year)))
plot(cv)
coef(cv, s="lambda.min")
```



```
> coef(cv, s="lambda.min")
7 x 1 sparse Matrix of class "dgCMatrix"
                      1
(Intercept) 2859.54398
T1            45.18485
T2          -133.90336
T3           -41.94147
RAD1          11.33300
RAD2          91.67204
RAD3          32.83682
```



Several issues

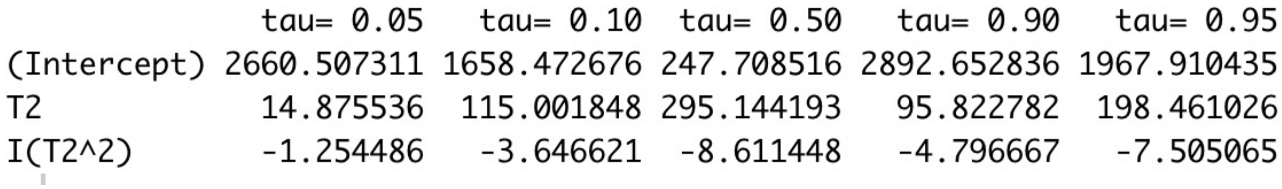
- Inputs X_1, …, X_P are sometimes (strongly) correlated
- Inputs X_1, …, X_P may have non-linear effects (unknown response shape),
- Too many inputs
- **Need to estimate extreme responses, not mean response**

Quantile regression

- Useful to estimate the response of *Y* to *X* for upper or lower quantiles of *Y*
- Relevant when using a limited number of inputs, i.e., one or two
- Useful for risk analysis
- Can be easily implemented with the R package quantreg

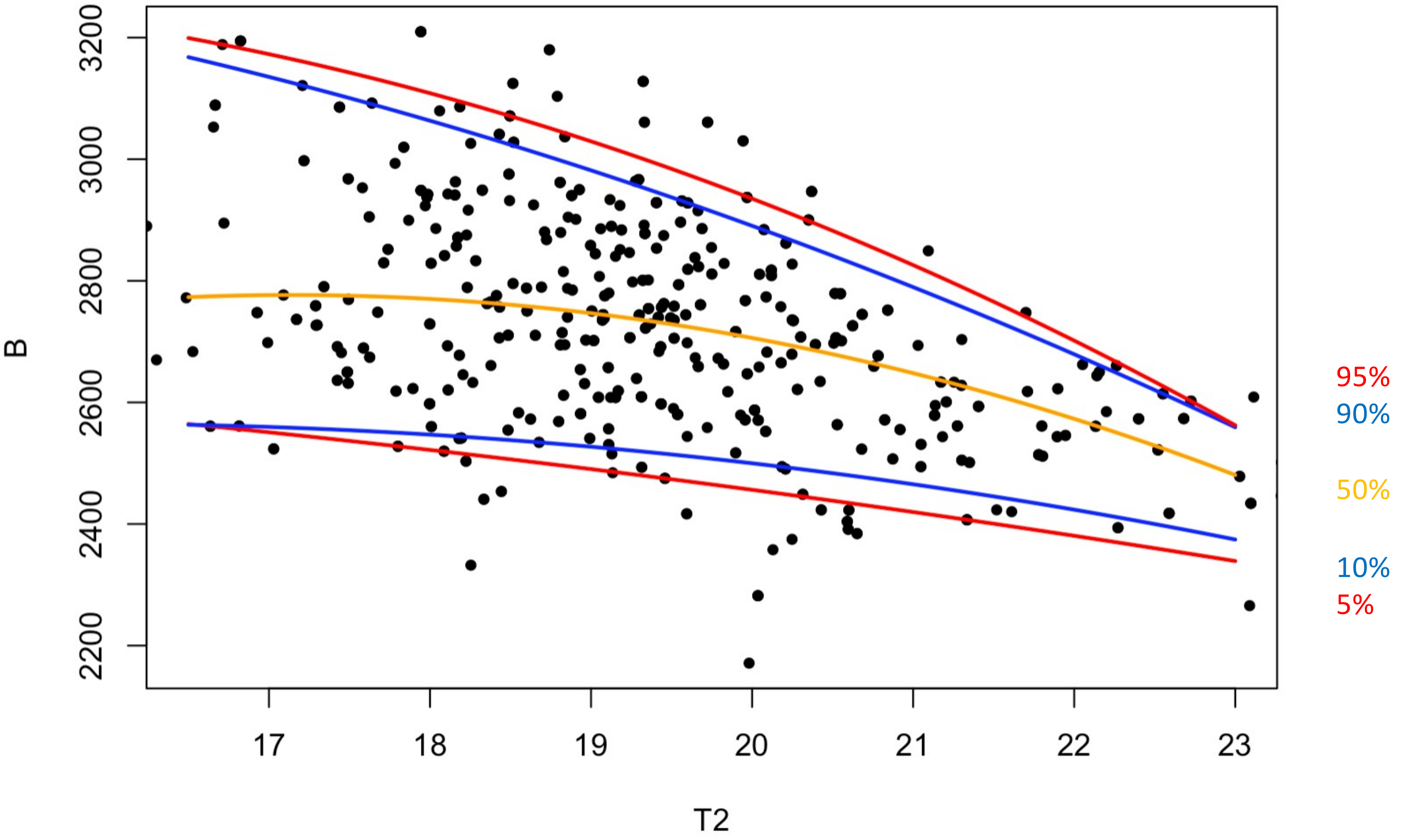
Example: maize biomass

```
library(quantreg)

mod<-rq(B~T2+I(T2^2), data=DataSet, tau=c(0.05,0.1, 0.5, 0.9, 0.95))
print(coef(mod))
```



```
               tau= 0.05   tau= 0.10  tau= 0.50   tau= 0.90   tau= 0.95
(Intercept) 2660.507311 1658.472676 247.708516 2892.652836 1967.910435
T2            14.875536  115.001848 295.144193   95.822782  198.461026
I(T2^2)       -1.254486   -3.646621  -8.611448   -4.796667   -7.505065
```

```
Xvec=seq(16.5,23,by=0.1)
Para=coef(mod)[,1]
lines(Xvec,Para[1]+Para[2]*Xvec+Para[3]*Xvec^2, col="red", lwd=2)

Para=coef(mod)[,2]
lines(Xvec,Para[1]+Para[2]*Xvec+Para[3]*Xvec^2, col="blue", lwd=2)

Para=coef(mod)[,3]
lines(Xvec,Para[1]+Para[2]*Xvec+Para[3]*Xvec^2, col="orange", lwd=2)

Para=coef(mod)[,4]
lines(Xvec,Para[1]+Para[2]*Xvec+Para[3]*Xvec^2, col="blue", lwd=2)

Para=coef(mod)[,5]
lines(Xvec,Para[1]+Para[2]*Xvec+Para[3]*Xvec^2, col="red", lwd=2)
```


3200
3000
2800
2600
2400
2200
B
17
18
19
20
21
22
23
T2
95%
90%
50%
10%
5%

Conclusion

- How to deal with strongly correlated inputs?

→ PCR, PLSR

- How to analyze the non linear relationships between Y and X_1, …, X_P?

→ GAM

- How to obtain accurate predictions with a simplified model?

→ Penalized regressions

- How to analyse responses at upper or lower quantiles?

→ Quantile regression