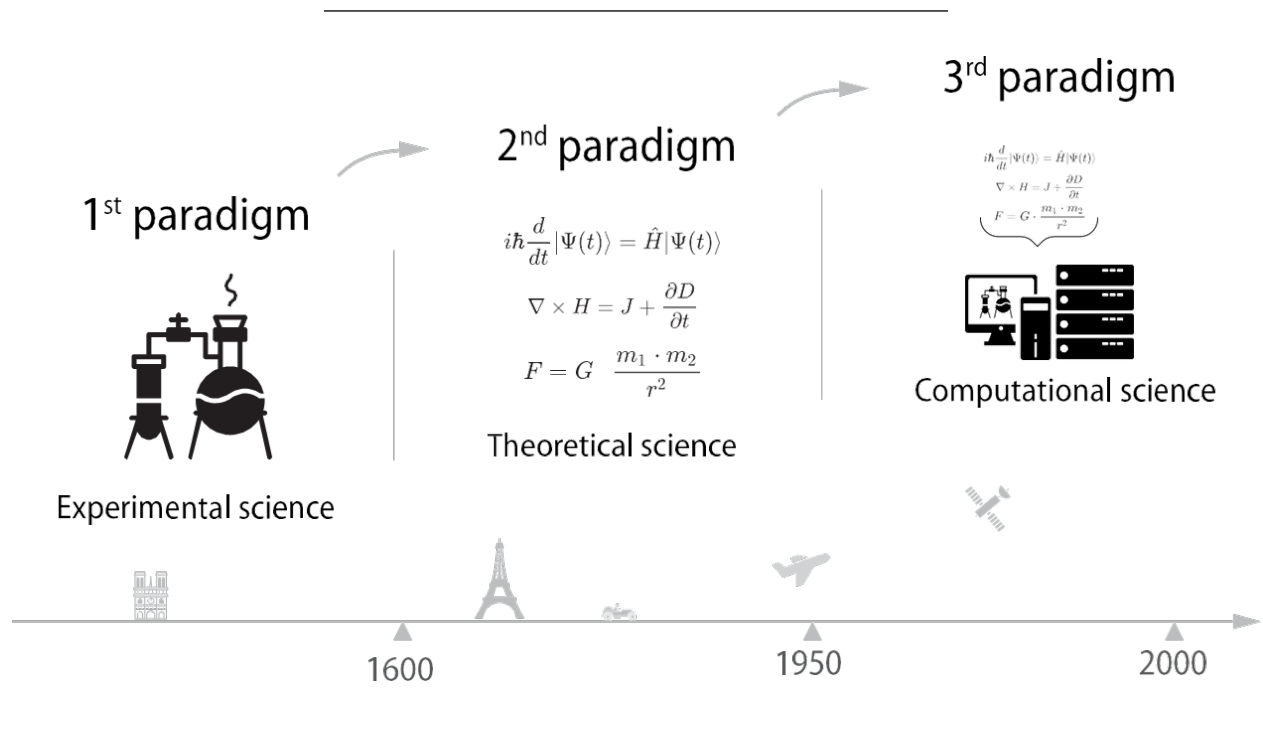
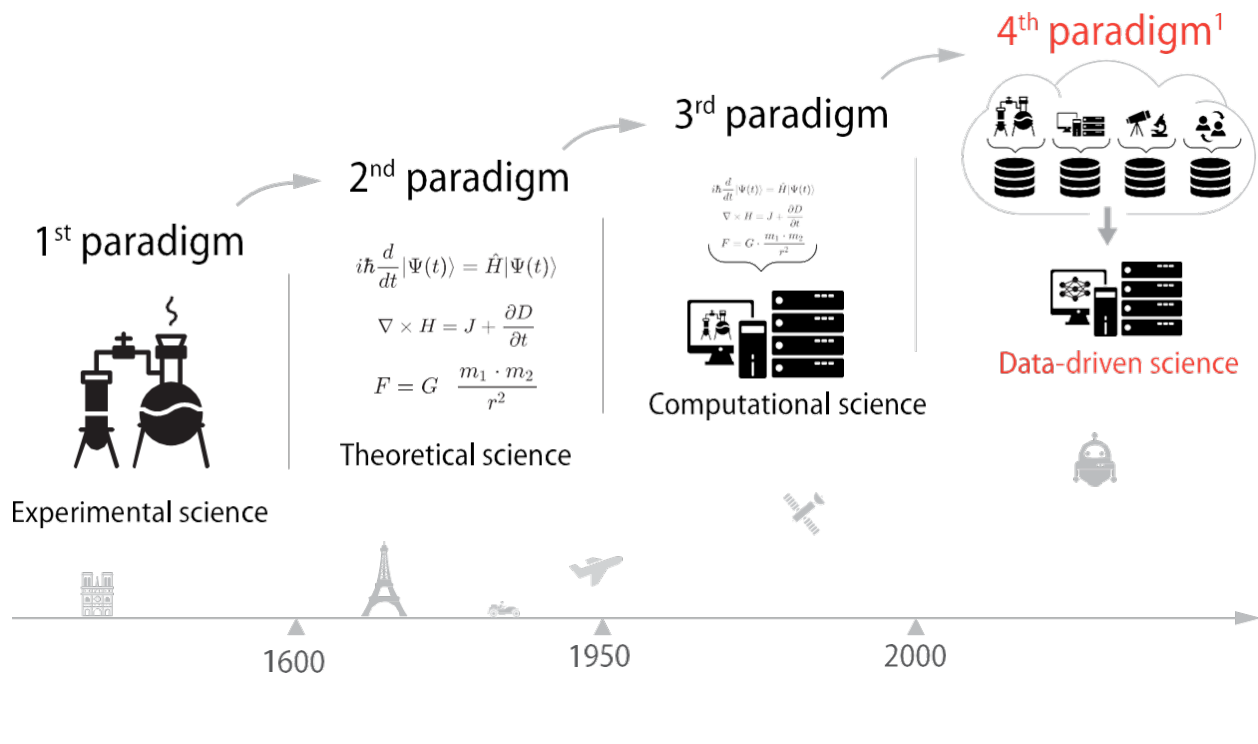


7-Machine learning

Thibaut FABACHER

Techniques prédictives/Apprentissage machine





Machine Learning

- Domaine de l'intelligence artificielle
- Consiste à entraîner des modèles informatiques à effectuer des tâches sans avoir été explicitement programmés pour les accomplir
- Les modèles peuvent s'améliorer au fil du temps en apprenant à partir de données
- Exemple : Traduction automatique Reconnaissance vocale...

Différence Stat / Machine learning

- décrire et comprendre les phénomènes à partir de données
- hypothetico-déductive (part d'hypothèses et utilise des tests statistiques pour les vérifier)
- données souvent de taille limitée et structurées
- modèles simples et faciles à comprendre
- prédire les résultats futurs à partir de données passées
- inductive (part de données et essaie de déduire les règles sous-jacentes)
- peut être utilisé avec des données de grande taille et non structurées
- modèles complexes et difficiles à interpréter (réseaux de neurones, arbres de décision)

01

Observation
empirique

« Il y a plus de cancer
chez les fumeurs »

02

Hypothèse

Le tabac est un facteur de
risque

03

Modèle

$K \sim \text{age} + \text{tabac} + \dots$

04

Résultat

Significativité de l'effet

01

02

03

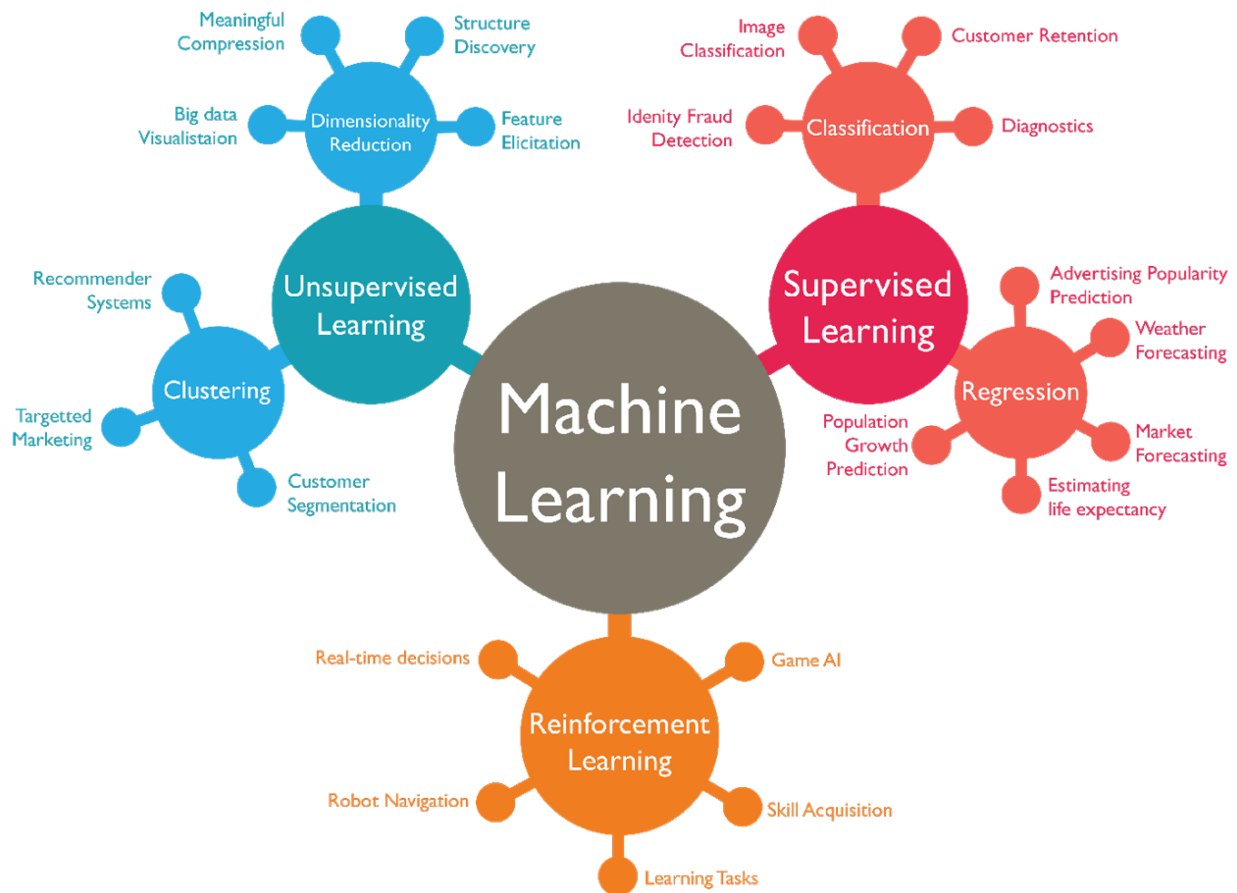
Modèle

$K \sim \text{age} + \text{tabac} + \dots$

04

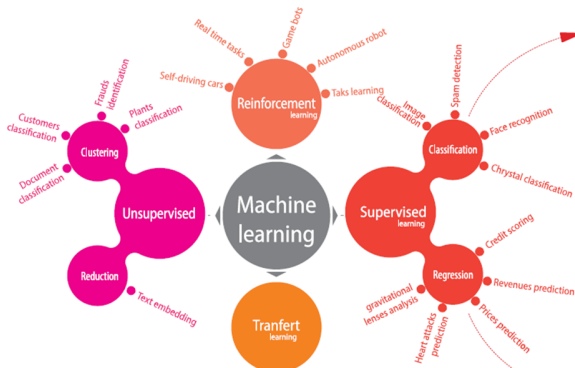
Résultat

Significativité de l'effet



Supervised learning

Learning from examples



Classification :

Predict qualitative informations



This is a cat



This is a rabbit



Tell me, what is it ?



Régression :

Predict quantitative informations



150 K€



400 K€



120 K€



100 K€



Tell me, what's the price ?



Apprentissage supervisé

- modèle est entraîné sur un jeu de données annotées
- Le jeu de données contient des exemples d'entrée et de sortie souhaités
- L'objectif: généraliser apprentissage à partir de ces exemples pour prédire la sortie correcte pour de nouvelles entrées

Apprentissage supervisé

$$\hat{y} = f(x, \theta)$$

où x est l'entrée, θ sont les paramètres du modèle et \hat{y} est la valeur prédite par le modèle pour l'entrée x .

Objectifs : Trouver les valeurs optimales de θ qui minimisent l'erreur entre les valeurs prédites et valeurs réelles.

- Fonction de coût avec optimisation

Apprentissage non supervisé

- Découvrir une structure au sein d'un ensemble d'individus caractérisés par des covariables X
- Label est inconnu

Apprentissage non supervisé

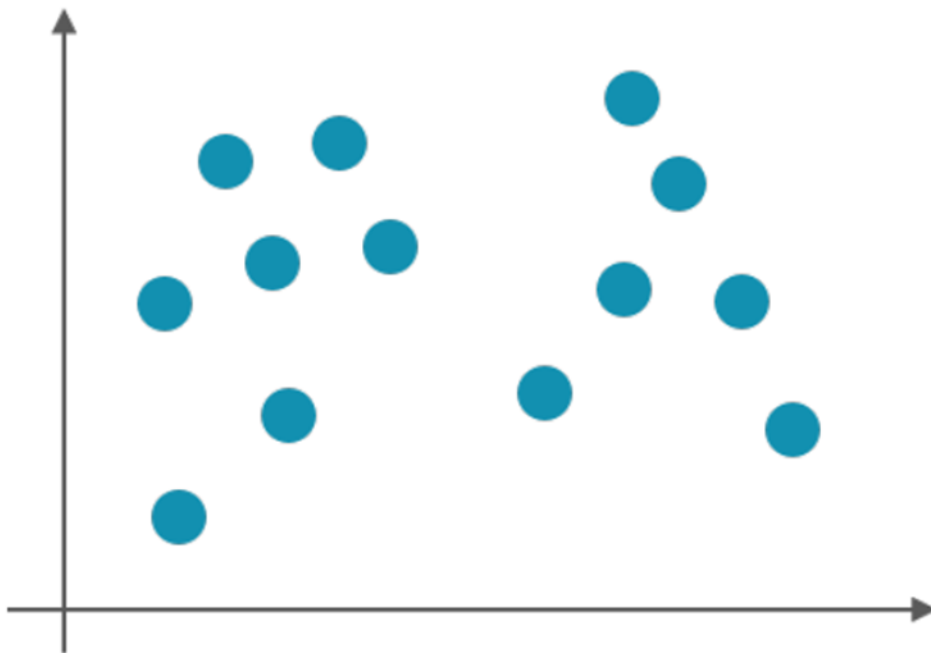
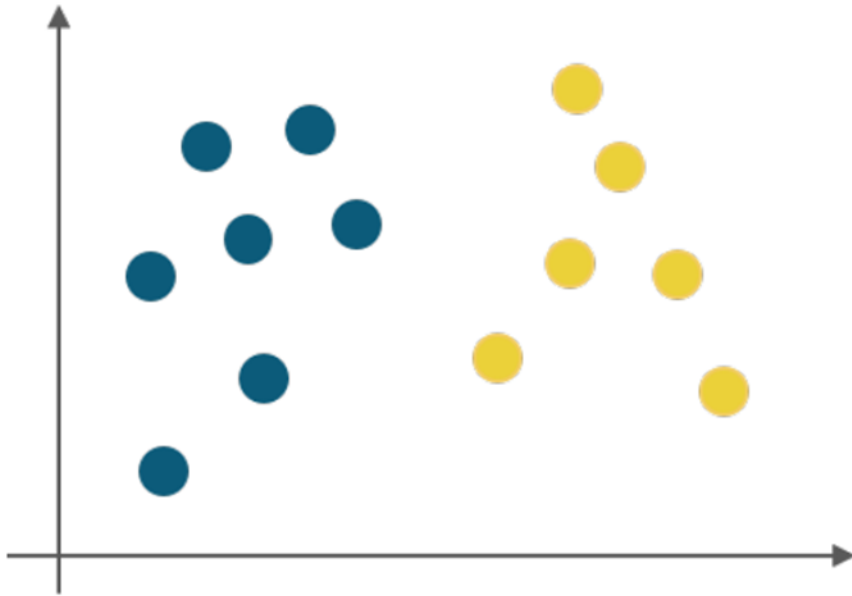
$$\hat{y} = f(x, \theta)$$

où x est l'entrée, θ sont les paramètres du modèle et \hat{y} est la valeur prédite par le modèle pour l'entrée x .

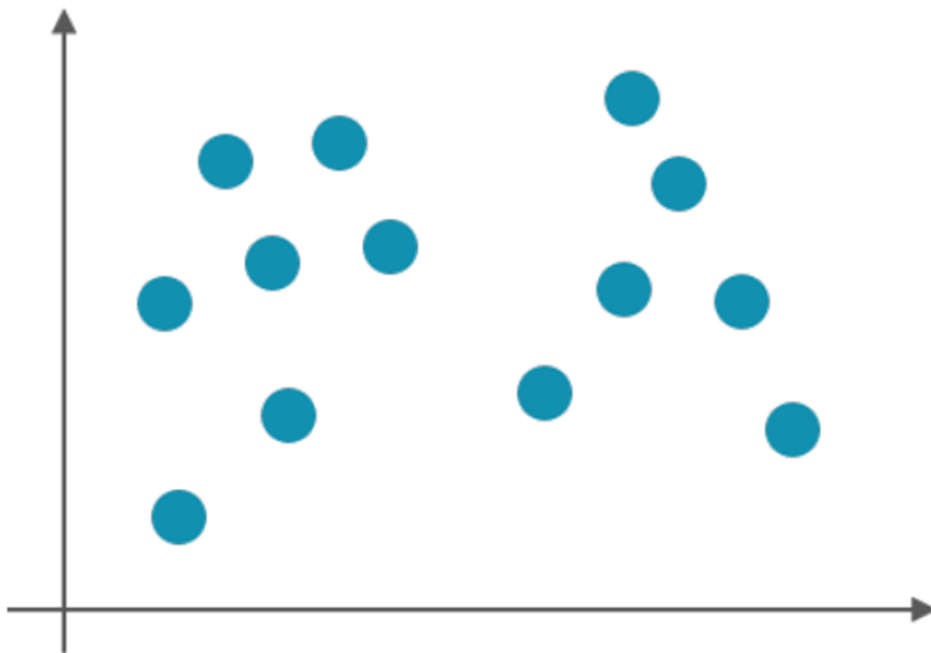
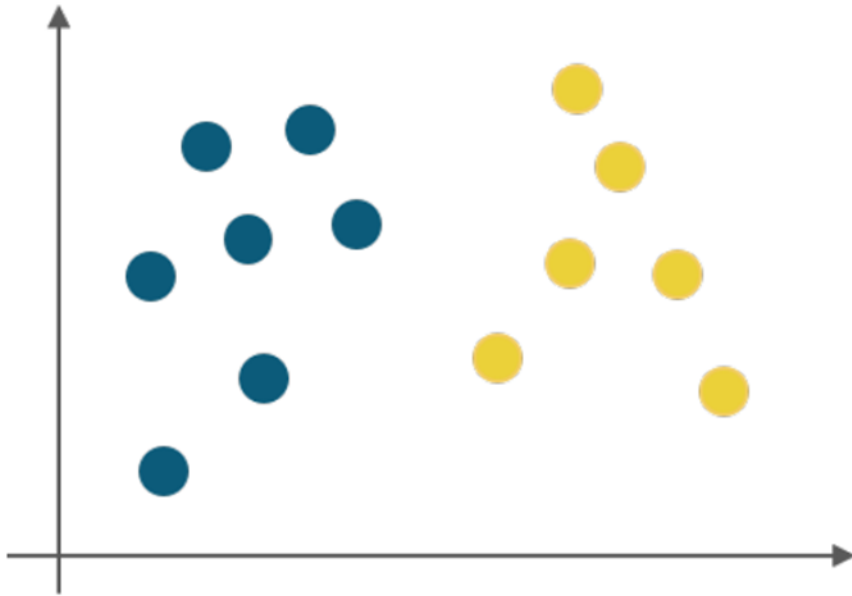
Objectifs : trouver des structures ou des patterns dans les données qui peuvent être utilisés pour effectuer des tâches utiles

Les paramètres du modèle sont mis à jour en utilisant une fonction de coût et une méthode d'optimisation afin de trouver des structures ou des patterns dans les données

Supervisé / Non supervisé



Supervisé / Non supervisé



Entrainement d'un modèle



Fonctions de coût

1. Erreur quadratique moyenne (MSE)

- Utilisée pour les tâches de régression
- Formule:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- Simple à calculer et à interpréter, mais sensible aux outliers et peu robuste face à la skewness des données

2. Erreur absolue moyenne (MAE)

- Utilisée pour les tâches de régression
- Formule:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

- Moins sensible aux outliers que le MSE, mais moins intuitive à interpréter
-

3. Erreur quadratique moyenne de racine (RMSE)

- Utilisée pour les tâches de régression
- Formule:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

- racine carrée de l'erreur quadratique moyenne (MSE)
 - échelle similaire aux vraies valeurs de sortie
-

4. Erreur de classification

- Utilisée pour les tâches de classification
- Formule:

$$Err_{class} = \frac{n_{erreurs}}{n}$$

- Simple à calculer, mais ne prend pas en compte la probabilité des prédictions
-

5. Indice de concordance de Cohen (Cohen's kappa)

- Utilisée pour les tâches de classification
- Formule:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

- moins sensible aux inégalités de distribution des classes que l'erreur de classification

Exemple sur la régression lineaire

```
## Chargement des données d'entraînement
train <- read.csv("train.csv")
X_train <- train[,1] ## variables explicatives
y_train <- train[,2] ## variable à prédire

## Entraînement du modèle de régression linéaire
model <- lm(y_train ~ X_train)

## Prédiction sur les données d'entraînement
y_pred <- predict(model, X_train)

## Calcul de l'erreur quadratique moyenne
mse <- mean((y_train - y_pred)^2)

## Affichage de l'erreur
print(mse)
```

Exemple sur données

```
## Load the serialized R object from the specified file
data_tot <- readRDS("data.rds")
```

```
## Set the seed for the random number generator to the value 45
set.seed(45)

## Generate a random sample of 100 elements from the rows of the 'data_tot' data frame, without replacement
sample_train <- sample(1:dim(data_tot)[1], 100, replace = F)

## Subset the 'data_tot' data frame to select only the rows in the random sample
data <- data_tot[sample_train,]
```

Création d'un jeu d'entraînement

```
## Fit a linear regression model to the data with 'hospital_los_day' as the dependent variable and 'sapsi_first' as the independent variable
fit1 <- lm(hospital_los_day ~ sapsi_first, data = data)

## Print a summary of the fitted model
summary(fit1)
```

Entraînement d'un modèle avec une variable

```
##
## Call:
## lm(formula = hospital_los_day ~ sapsi_first, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.354 -5.782 -2.486  2.186 53.259
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.0248     3.4826   2.304  0.0233 *
## sapsi_first    0.1226     0.2234   0.549  0.5844
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.061 on 98 degrees of freedom
## Multiple R-squared:  0.003063,    Adjusted R-squared:  -0.00711
## F-statistic: 0.3011 on 1 and 98 DF,  p-value: 0.5844
```

```
## Use the fitted model to make predictions on the data
y_pred <- predict(fit1, data)

## Calculate the mean squared error between the actual dependent variable values and the predicted values
mse <- mean((data$hospital_los_day - y_pred)^2)

print(mse)
```

Calcul de la MSE

```
## [1] 80.46587
```

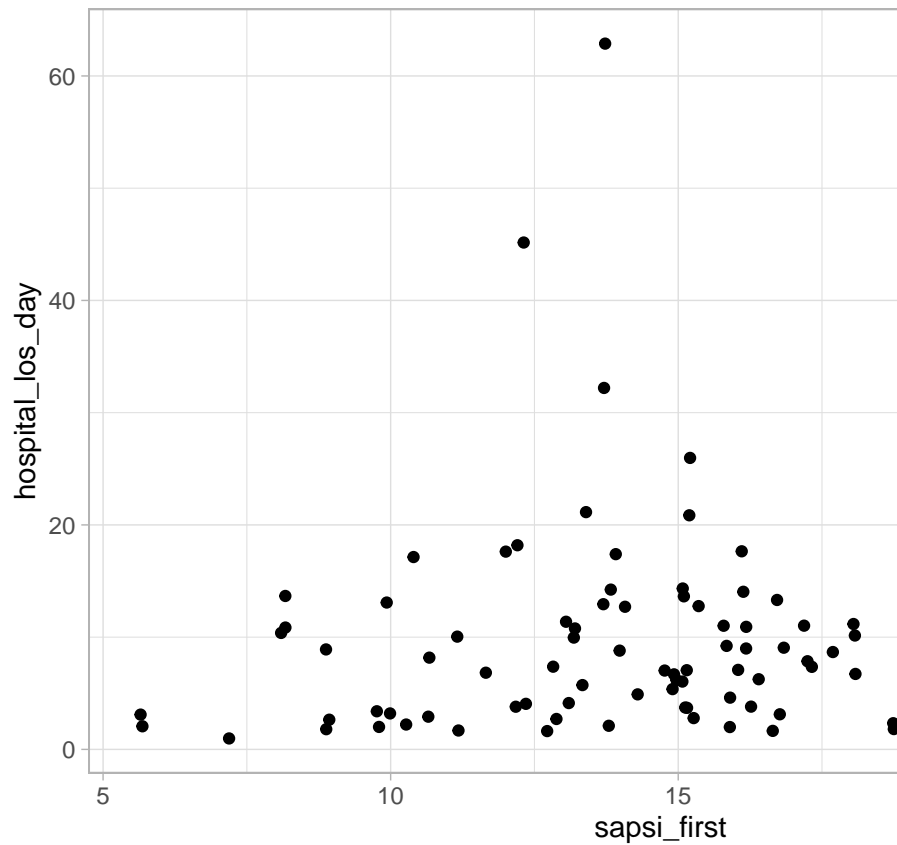
```
#### Add second variable
fit2 <- lm(hospital_los_day ~ sapsi_first+age, data = data)
##summary(fit2)
y_pred2 <- predict(fit2, data)
mse2 <- mean((data$hospital_los_day - y_pred2)^2)

print(mse2)
```

Modèle avec deux variable

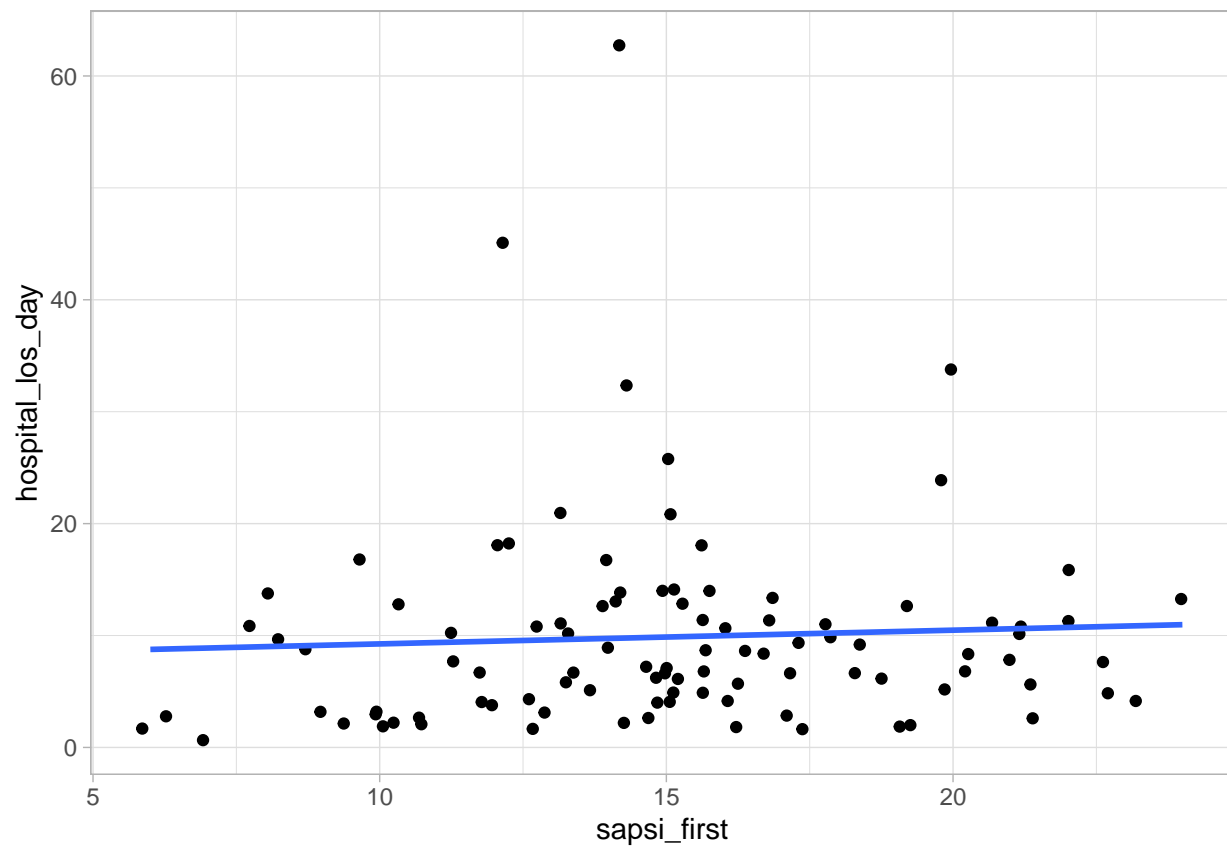
```
## [1] 79.89498
```

```
p1<-data %>% ggplot(aes(x=sapsi_first,y=hospital_los_day))+geom_jitter()+theme_light()
p1
```

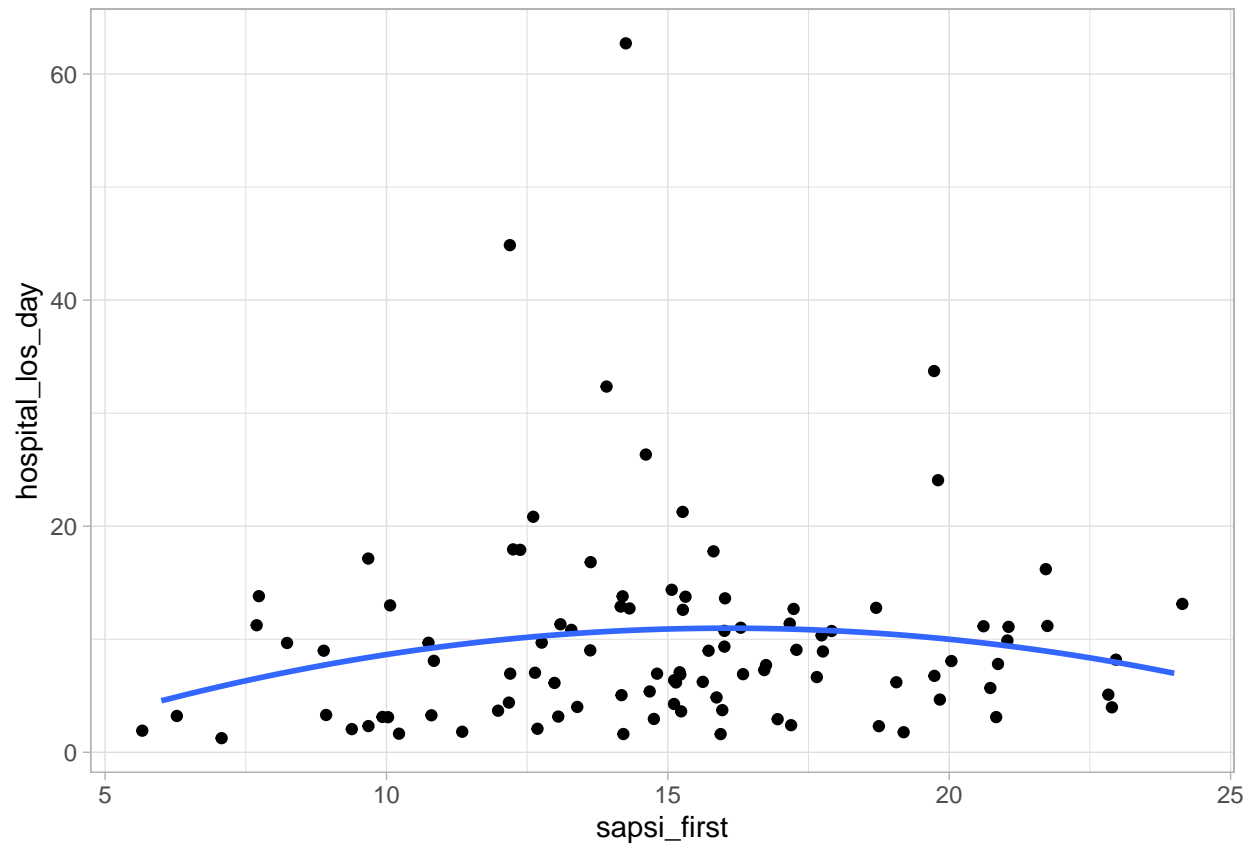


Autre type de choix d'hyperparamètres

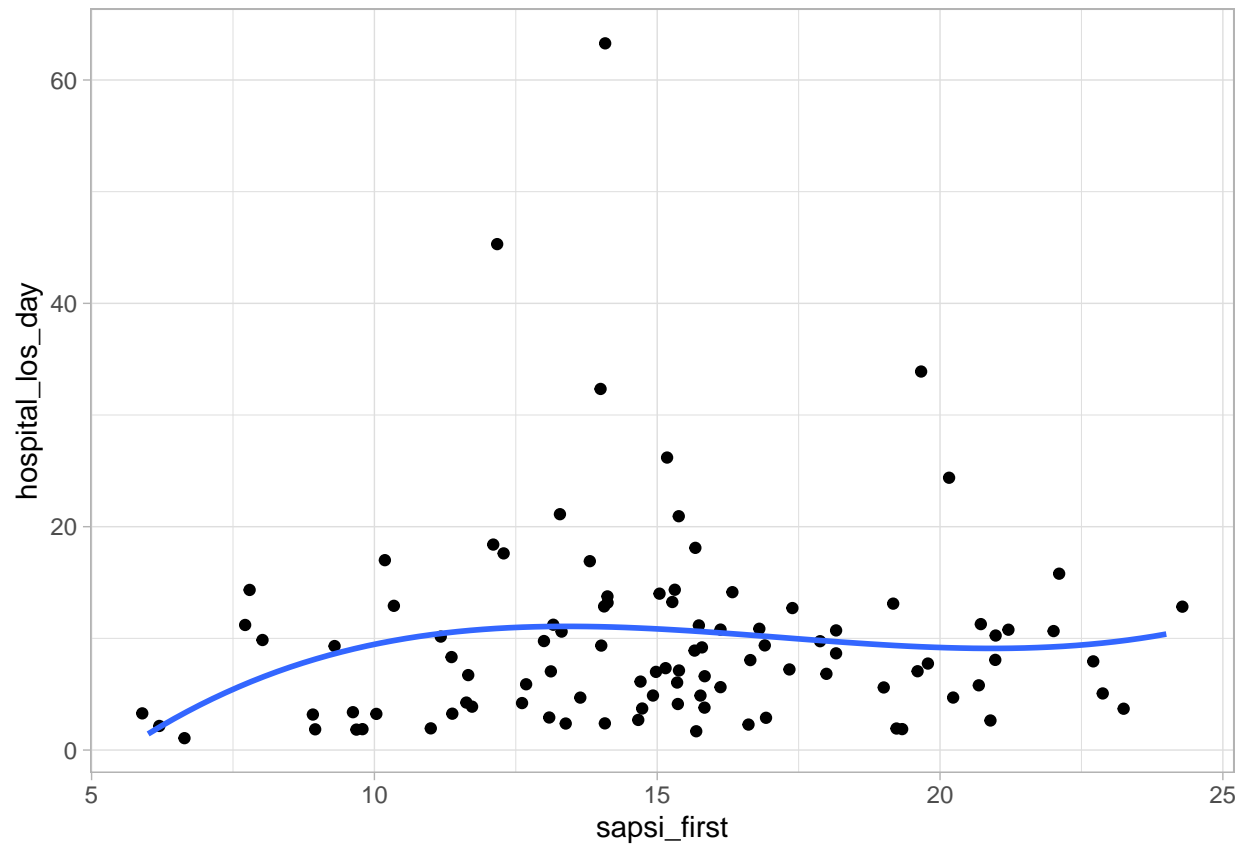
```
p2<-data %>% ggplot(aes(x=sapsi_first,y=hospital_los_day))+geom_jitter()+theme_light()+
  geom_smooth( method = lm, formula = y ~ x, se = FALSE)
p2
```



```
p3<-data %>% ggplot(aes(x=sapsi_first,y=hospital_los_day))+geom_jitter()+theme_light()+  
  geom_smooth( method = lm, formula = y ~ poly(x,2), se = FALSE)  
p3
```



```
p4<-data %>% ggplot(aes(x=sapsi_first,y=hospital_los_day))+geom_jitter()+theme_light()+  
  geom_smooth( method = lm, formula = y ~ poly(x,3), se = FALSE)  
p4
```



```
library(gridExtra)
```

```
##  
## Attachement du package : 'gridExtra'  
  
## L'objet suivant est masqué depuis 'package:dplyr':  
##  
##      combine
```

```
grid.arrange(p1, p2, p3, p4, nrow = 2)
```

