

# Sélection de variables

## Analyse multidimensionnelle appliquée

Léo Belzile

HEC Montréal

Objectif: bâtir un modèle pour une variable réponse  $Y$  en fonction de variables explicatives  $X_1, \dots, X_p$ .

On s'intéresse à

$$f(X_1, \dots, X_p) \quad .$$

vraie moyenne inconnue

L'analyste détermine

$$\hat{f}(X_1, \dots, X_p),$$

approximation

une fonction des variables explicatives.

On spécifie que la moyenne de la variable réponse  $Y$  est une fonction linéaire des variables explicatives  $X_1, \dots, X_p$ , soit

$$\underset{\text{moyenne théorique}}{E(Y \mid X)} = \underset{\text{somme pondérée des variables explicatives}}{\beta_0 + \beta_1 X_{i1} + \dots + \beta_p X_{ip}} .$$

en supposant que l'écart entre les observations et cette moyenne est constant,

$$\text{Va}(Y \mid X) = \sigma^2.$$

Pour la  $i$ ie observation,

$$\underset{\text{réponse}}{Y_i} = \beta_0 + \beta_1 X_{i1} + \dots + \beta_p X_{ip} + \underset{\text{aléa}}{\varepsilon_i}.$$

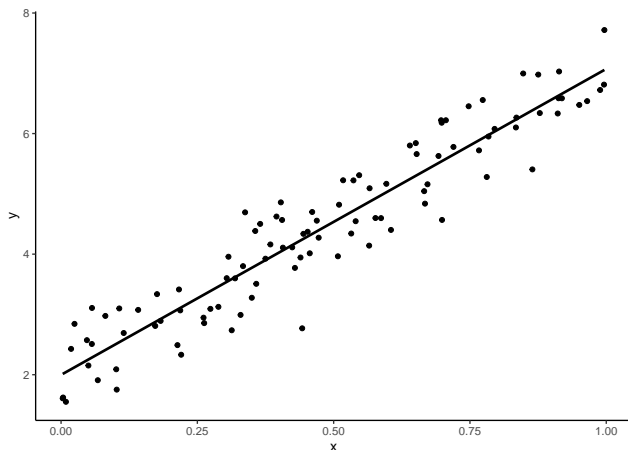
- L'aléa  $\varepsilon_i$  représente la distance verticale entre la vraie pente et l'observation
- Autant d'aléas que d'observations ( $n$ ), variable aléatoire inconnue...

- L'aléa  $\varepsilon_i$  représente l'erreur, soit la différence entre la valeur observée et la moyenne de la population pour les même valeurs des variables explicatives.
- On suppose que le modèle pour la moyenne est correctement spécifié: l'aléa a une moyenne théorique nulle,  $E(\varepsilon_i) = 0$ .
- On suppose que les observations sont indépendantes les unes des autres.

# Régression linéaire en deux dimensions

Si  $E(Y) = \beta_0 + \beta_1 X$ , alors

- $\beta_0$  représente l'ordonnée à l'origine (valeur quand  $X = 0$ .)
- $\beta_1$  est la pente



L'estimation des paramètres  $\hat{\beta}_0, \dots, \hat{\beta}_p$  nous donne

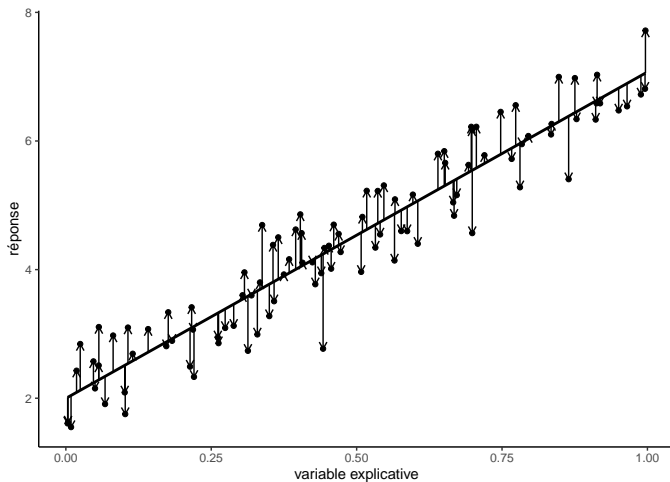
$$\underset{\text{prédiction}}{\hat{Y}_i} = \hat{\beta}_0 + \hat{\beta}_1 X_{i1} \dots + \hat{\beta}_p X_{ip}.$$

On peut approximer l'aléa à l'aide du résidu ordinaire, soit

$$\underset{\text{résidu ordinaire}}{e_i} = \underset{\text{observation}}{Y_i} - \underset{\text{prédiction}}{\hat{Y}_i}.$$

- par construction, la moyenne des  $e_i$  est zéro.
- le résidu ordinaire est la distance verticale entre l'observation et la "droite" ajustée

# Illustration des résidus ordinaires





L'erreur quadratique moyenne théorique est

$$E \left[ \left\{ Y - \hat{f}(x_1, \dots, x_p) \right\}^2 \right],$$

la moyenne de la différence au carré entre la vraie valeur de  $Y$  et la valeur prédite par le modèle.

En pratique, on remplace la moyenne théorique par une moyenne empirique obtenue à partir d'un échantillon aléatoire.

Comment estimer les paramètres  $\beta_0, \dots, \beta_p$ ?

Optimisation: trouver les valeurs qui minimisent l'erreur quadratique moyenne empirique avec l'échantillon des  $n$  observations, soit

$$\frac{e_1^2 + \dots + e_n^2}{n}$$

Il existe une solution explicite au problème d'optimisation!

La fonction `lm` calcule l'ajustement du modèle linéaire.

Arguments:

- `formula`: formule de type `reponse ~ variables explicatives`, où les variables explicatives sont séparées par un signe `+`
- `data`: base de données

```
1 modlin <- lm(mpg ~ hp + wt,  
2             data = mtcars)  
3 summary(modlin)
```

# Sortie de summary

Call:

```
lm(formula = mpg ~ hp + wt, data = mtcars)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.941	-1.600	-0.182	1.050	5.854

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	37.22727	1.59879	23.285	< 2e-16 ***
hp	-0.03177	0.00903	-3.519	0.00145 **
wt	-3.87783	0.63273	-6.129	1.12e-06 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.593 on 29 degrees of freedom

Multiple R-squared: 0.8268, Adjusted R-squared: 0.8148

F-statistic: 69.21 on 2 and 29 DF, p-value: 9.109e-12

- Formule de l'appel
- Statistiques descriptives des résidus ordinaires  $e_1, \dots, e_n$ .
- Tableau des estimations
  - Coefficients  $\hat{\beta}_j$
  - Erreurs-types,  $se(\hat{\beta}_j)$
  - Statistique du test-t pour  $\mathcal{H}_0 : \beta_j = 0$ , soit  $t = \hat{\beta}_j / se(\hat{\beta}_j)$
  - Valeur-p selon loi nulle  $St(n - p - 1)$
- Estimation de l'écart-type  $\hat{\sigma}$  et degrés de liberté  $n - p - 1$
- Estimations du coefficient de détermination,  $R^2$  et  $R^2$  ajusté
- Statistique  $F$  d'ajustement global et valeur- $p$  de  $F(p, n - p - 1)$ 
  - $\mathcal{H}_a$ : modèle linéaire
  - $\mathcal{H}_0$ : modèle avec uniquement ordonnée à l'origine (chaque observation prédite par la moyenne des réponses,  $\bar{Y}$ )

- `resid` pour les résidus ordinaires  $e_i$
- `fitted` pour les valeurs ajustées  $\hat{Y}_i$
- `coef` pour les estimations des paramètres  $\hat{\beta}_0, \dots, \hat{\beta}_p$
- `plot` pour des diagnostics graphiques d'ajustement
- `anova` pour la comparaison de modèles emboîtés
- `predict` pour les prédictions (avec nouvelles données)
- `confint` pour intervalles de confiance pour les paramètres.

- Les facteurs (`<factor>`) sont traités adéquatement par R.
- Si la variable a  $K$  valeurs possibles (niveaux), le modèle inclut  $K - 1$  indicatrices 0/1.
- Par défaut dans R, la catégorie de référence est la plus petite en ordre alphanumérique.

Considérons une variable catégorielle `cat` avec niveaux 1, 2, et 3.

cat	cat2	cat3
1	0	0
2	1	0
3	0	1

La catégorie de référence est associée à l'ordonnée à l'origine (quand `cat2=0` et `cat3=0`).



- Comment choisir quelles variables inclure?
- Quel est la spécification adéquate pour  $f(X_1, \dots, X_p)$ ?
  - régression, réseaux de neurone, forêts aléatoires, etc.
  - transformations de variables,  $\text{age}^2$ ,  $\ln(\text{age})$ , etc.

Notre but sera de sélectionner un bon modèle, selon les objectifs de l'étude

Prédiction: obtenir une estimation de  $\hat{Y}$ : on veut un modèle performant

Inférence: estimer l'effet de variables explicatives, effectuer des tests d'hypothèses

Pour l'inférence, il est préférable de spécifier le modèle dès le départ (devis expérimental) selon des considérations scientifiques et de s'y tenir.

## Démonstration R

Omettre des termes importants mène à un modèle biaisé.  
Ajouter des termes superflus augmente la variabilité (on estime des zéros).

On peut calculer l'erreur quadratique moyenne (EQM) sur l'ensemble des données qui servent à ajuster le modèle.

Q: Est-ce que c'est un marqueur fiable de la performance du modèle?

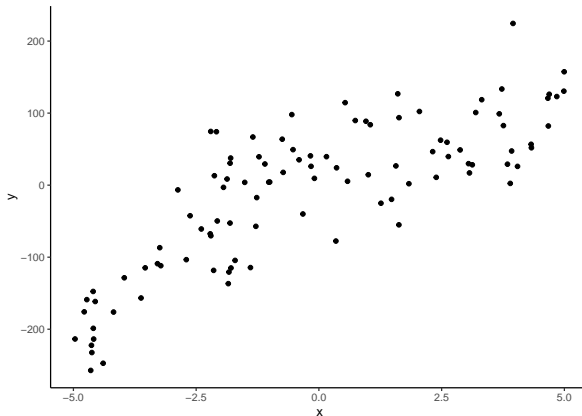
# Exemple avec la régression polynomiale

On a simulé des observations d'un modèle polynomiale d'ordre  $K$ , de la forme

$$E(Y \mid X) = \beta_0 + \beta_1 X + \dots + \beta_K X^K.$$

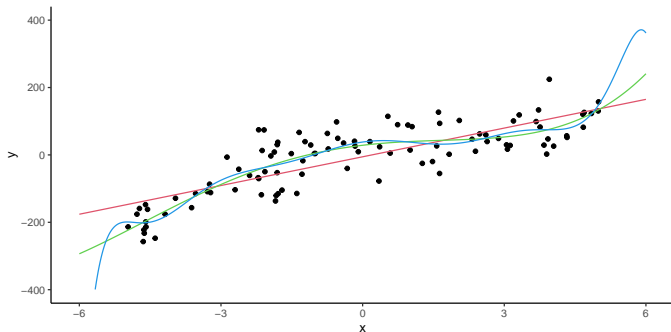
On vise à estimer l'ordre du polynôme en effectuant des régressions linéaires.

```
1 data(polynome, package = "hecmulti")
2 k <- 4L # degré du polynôme
3 lm(y ~ poly(x, degree = k),
4     data = polynome)
```



D'après vous, quelle est la vraie valeur de  $K$ ?

# Ajustement de polynômes



Ajustement pour des polynômes de degré  $K = 1, 4, 10$ .

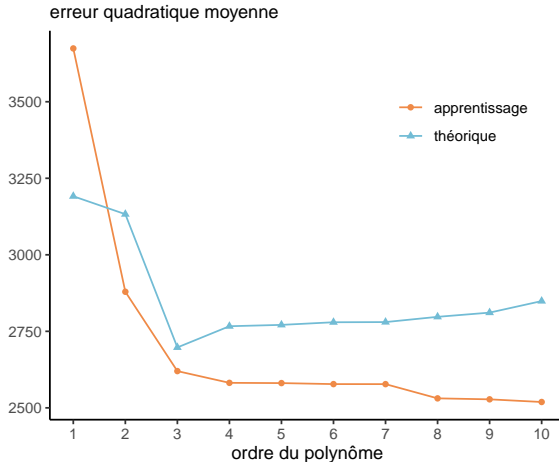
Pour  $K = 1, \dots, 10$ :

- Ajuster le modèle linéaire
- Obtenir les résidus ordinaires
- Calculer l'erreur quadratique moyenne

```
1 data(polynome, package = "hecmulti")
2 eqm <- vector(length = 10L, mode = "numeric")
3 for(K in seq_len(10)){
4   mod <- lm(y ~ poly(x, degree = K), data = polynome)
5   eqm[K] <- mean(resid(mod)^2)
6 }
```



# Estimation de l'erreur quadratique moyenne



Plus le modèle est complexe, plus l'erreur quadratique moyenne de l'échantillon d'apprentissage est petite!

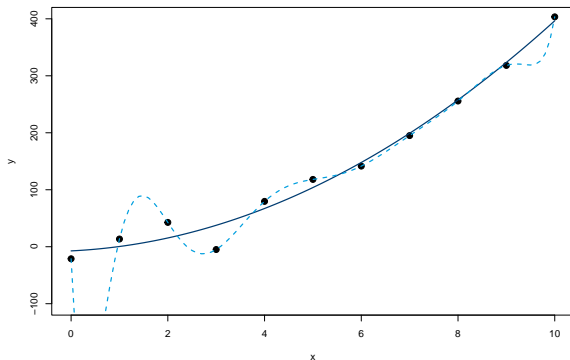
L'erreur quadratique moyenne décroît mécaniquement à chaque fois qu'on ajoute une variable au modèle de régression.

C'est pourquoi on ne peut pas l'utiliser comme outil de sélection de variables, autrement on va surestimer la performance (suroptimisme).

# Surajustement

Un modèle plus complexe va toujours mieux s'ajuster aux données de l'échantillon.

En revanche, il se généralise moins bien (notre objectif en prédiction).



Nous nous sommes rendus coupables d'un péché capital en utilisant deux fois les données

- une fois pour l'ajustement,
- une fois pour la validation.

C'est comme tremper deux fois un biscuit dans le verre de lait...

- Pénalisation de la complexité du modèle
  - critères d'information
  - pénalité sur les coefficients (ridge, LASSO)
- Validation et entraînement avec des données différentes
  - validation externe
  - validation croisée

Utiliser toutes les données (échantillon de validation), mais ajouter une pénalité.

Si le modèle est ajusté avec la méthode du maximum de vraisemblance, alors on a accès aux critères d'information.

Si on suppose que les aléas sont indépendants et suivent une loi normale de variance  $\sigma^2$  constante, alors la log-vraisemblance s'écrit

$$\begin{aligned}\ell(Y; \mathbf{X}, \beta, \sigma) = & -\frac{n}{2} \ln(2\pi) - n \ln(\sigma) \\ & - \frac{1}{2\sigma^2} \sum_{i=1}^n (Y_i - \beta_0 - \dots - \mathbf{x}_{ip}\beta_p)^2.\end{aligned}$$

Ainsi, l'estimateur des moindres carrés,  $\hat{\beta}$ , correspond au maximum de vraisemblance des coefficients.

Valables uniquement pour les modèles ajustés par maximum de vraisemblance, ils sont de la forme

$$IC = -\text{ajustement} + \text{pénalité} \times \text{nb param}$$

- On veut maximiser l'ajustement, donc minimiser le premier terme.
- La pénalité vient contrebalancer l'amélioration mécanique de l'ajustement quand le modèle a plus de flexibilité.
- Plus le critère d'information est petit, meilleur c'est.



Pour la régression linéaire, on peut réécrire les critères d'information sous la forme

$$AIC = n \ln(\widehat{EQM}) + 2 \times \text{nb param} + \text{constante},$$

$$AIC = n \ln(\widehat{EQM}) + \ln(n) \times \text{nb param} + \text{constante},$$

- Le critère d'Akaike (AIC) utilise une pénalité de 2, le critère bayésien (BIC) de  $\ln(n)$ .
- Le BIC (critère Bayésien de Schwarz) pénalise donc davantage que le AIC (critère d'Akaike).

- Plus la valeur du critère d'information est petite, meilleure est l'adéquation (et donc la performance).
- La pénalité assure que les modèles plus complexes ne sont pas systématiquement retournés.
- Le BIC retourne toujours des modèles plus parcimonieux (c'est-à-dire avec moins de paramètres) que le AIC.
- Génériques AIC et BIC en R

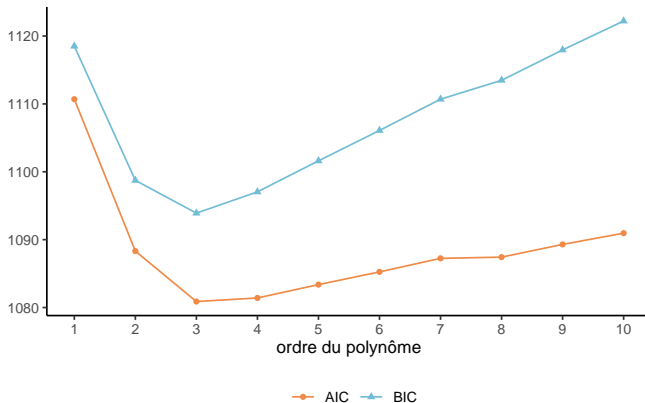


Figure 1: Critères d'information en fonction de l'ordre du polynôme.

Les deux critères d'information retournent  $K = 3$  comme choix optimal.

Ne pas utiliser les données employés pour ajuster un modèle pour prédire la performance.

- échantillons d'apprentissage/validation/test (fixes)
- validation croisée

Séparer l'ensemble d'observations en plusieurs groupes

- l'échantillon d'apprentissage pour l'ajustement du modèle
- l'échantillon de validation pour l'estimation de la performance
- l'échantillon test pour l'inférence (optionnel)

En pratique, il faut beaucoup d'observations!

Applicable en toute généralité peu importe le type de modèle.

Cette approche, qui compartimente les échantillons, n'est pas sans faille.

- On obtient un résultat différent selon la division
- Gaspillage potentiel
- Choix d'ordinaire aléatoire, mais choix particuliers de fenêtre selon données (par ex., séries chronologiques)

Une autre méthode de rééchantillonnage

Diviser l'échantillon en  $M$  groupes d'observations de taille moyenne égale

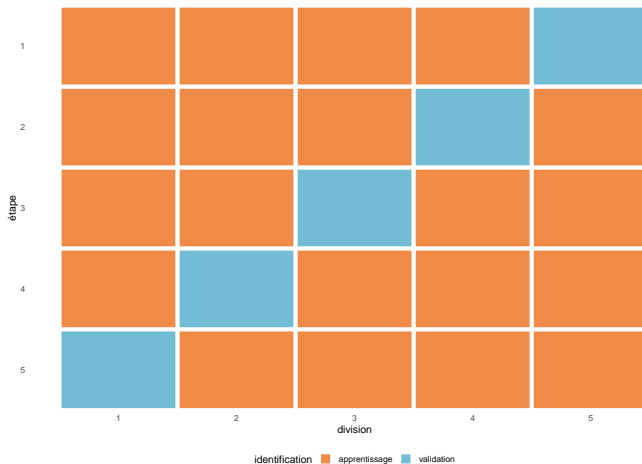
- utiliser  $M - 1$  groupes pour l'ajustement
- estimer la performance avec les données du dernier groupe

Les regroupements sont aléatoires, tout comme la mesure finale de performance!

1. Diviser l'échantillon au hasard en  $M$  parties  $P_1, P_2, \dots, P_M$  contenant toutes à peu près le même nombre d'observations.
2. Pour  $j = 1$  à  $M$ ,
  - i. Enlever la partie  $j$ .
  - ii. Estimer les paramètres du modèle en utilisant les observations des  $M - 1$  autres parties combinées.
  - iii. Calculer la mesure de performance (par exemple la somme du carré des erreurs  $\{Y_i - \hat{Y}_i\}^2$ ) de ce modèle pour le groupe  $P_j$ .
3. Combiner les  $M$  estimations de performance pour obtenir une mesure de performance finale.



# Illustration de la validation croisée



# Combien de groupes $M$ pour la validation croisée?

La validation croisée est plus coûteuse parce qu'on doit ajuster  $M$  fois le modèles

- Pour la régression linéaire, choisir  $M = n$  (leave-one-out cross-validation) permet d'éviter de réajuster le modèle grâce à un artifice de calcul.
- On recommande habituellement de prendre  $\min\{n^{1/2}, 10\}$  groupes,
- mais les choix  $M = 5$  ou  $M = 10$  sont ceux qui revient le plus souvent en pratique.

Voir Davison & Hinkley (1997), Bootstrap methods and their applications, Section 6.4 pour une discussion plus étoffée et l'algorithme 6.5 pour une meilleure implémentation.

# Validation croisée = résultat aléatoire!

Soit  $\widehat{\text{EQM}}_{\text{VC}}$  l'estimation de l'erreur quadratique moyenne obtenue par validation croisée à  $M$  plis pour un modèle  $\mathcal{M}$  donné.

Si on a un nombre similaire d'observations dans chaque groupe:

- calculer l'erreur quadratique moyenne de chaque pli,  $\widehat{\text{EQM}}_{\text{VC},m}$
- si  $M$  est suffisamment grand (disons  $M \geq 10$ ), on peut estimer l'écart-type empirique de ces moyennes

$$\text{sd} = \frac{1}{M-1} \sum_{m=1}^M (\widehat{\text{EQM}}_{\text{VC},m} - \widehat{\text{EQM}}_{\text{VC}})^2$$

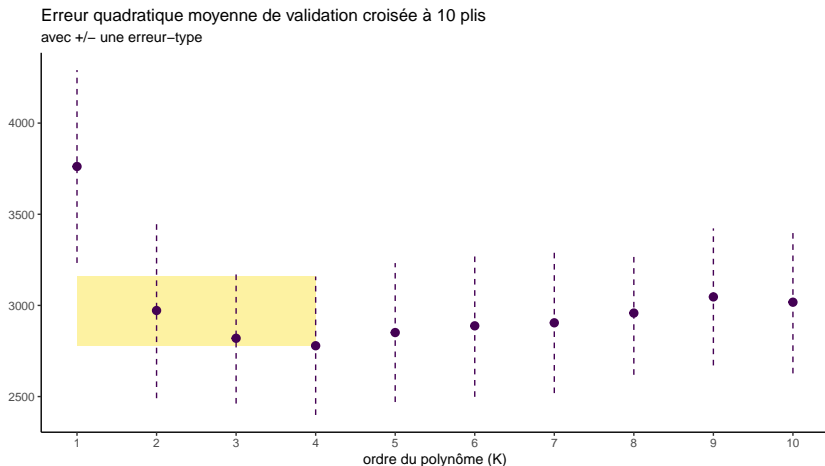
- Pour obtenir l'erreur-type de la moyenne globale, diviser par la racine de la taille du nombre de groupes, d'où

$$\text{se}(\widehat{\text{EQM}}_{\text{VC}}) = \text{sd} / \sqrt{M}$$

Soit  $\mathcal{M}_{\text{opt}}$  le modèle avec la plus petite erreur quadratique moyenne de validation croisée.

Suivant Breiman, on choisit le modèle le plus simple qui soit à au plus une erreur-type de la performance du modèle  $\mathcal{M}_{\text{opt}}$ , en prenant  $\text{se}\{\widehat{\text{EQM}}_{\text{VC}}(\mathcal{M}_{\text{opt}})\}$ .

# Illustration de la règle de une erreur-type



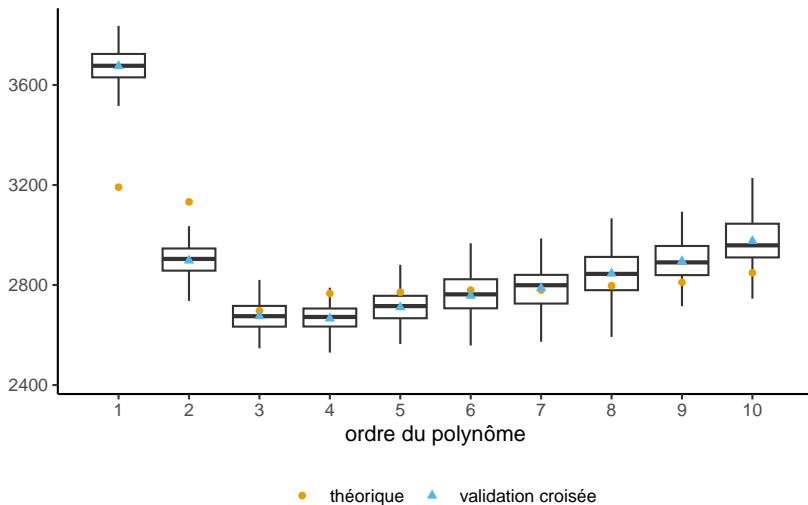
La bande jaune donne une erreur-type du meilleur modèle. On choisirait le modèle avec  $M = 2$ , plutôt que  $M = 4$

Le paquet `caret` a une fonction pour faire la validation croisée.

```
1 cv_caret <-  
2   caret::train(form = formula(y ~ poly(x, degree = 3)),  
3               data = polynome,  
4               method = "lm",  
5               trControl = caret::trainControl(  
6                 method = "cv",  
7                 number = 10)) #nb plis  
8 reqm_cv <- cv_caret$results$RMSE # racine EQM  
9 reqm_se_cv <- cv_caret$results$RMSESD / sqrt(10)  
10 # calcul de l'erreur-type de la racine EMQ pour ce modèle
```

Aussi `boot::cv.glm()` qui inclut une correction de biais pour les modèles linéaires généralisé.

# Résultats de la validation croisée



- Le choix de la complexité d'un modèle est un compromis entre
  - le biais (modèle trop simple, mal spécifié) et
  - la variance (même nombre d'observations/budget, plus de paramètres à estimer, estimations moins fiables)
- L'erreur quadratique moyenne est la mesure usuelle de la performance d'un modèle linéaire



Si on estime la performance avec les mêmes données qui ont servi à l'ajustement, on surestime la performance

- l'erreur quadratique moyenne calculée sur les mêmes données qui ont servi à l'entraînement est biaisée
- cela mène à du surajustement

Trois méthodes pour estimer de manière plus objective la performance d'un modèle

- Critères d'information (pénalisation)
- Validation externe
- Validation croisée

Vous devez être en mesure de nommer les forces et faiblesses et d'expliquer le fonctionnement (avec du pseudocode).