

# Éléments pour Revealjs

Marc Thévenin

Institut National d'Etudes Démographiques

2024-03-20

- Documentation officielle:
  - [La présentation](#)
  - [Le code source](#)
- [Code source de cette présentation](#)

# Report des Codes

# Affichage des codes

- Par défaut les blocs ne sont pas affichés:
  - Prévoir dans le yaml: soit **echo: true** ou **code-fold: true**.
- Jouer avec ces deux options dans chaque bloc individuellement.

# Line highlight

- Option `code-line-numbers`: "valeurs lignes"<sup>1</sup>
  - De préférence en option du corp du bloc: `#| code-line-numbers: "valeurs lignes"`
  - Exemples:
    - `code-line-numbers: "1"` [ligne 1]
    - `code-line-numbers: "1,4"` [ligne 2 et 4]
    - `code-line-numbers: "1-4"` [ligne 2 à 4]
    - `code-line-numbers: "|2|4"` [toutes les lignes, puis 2, puis 4]

1. Cette option ne fonctionne pas chez avec certains thès: exemple le thème *league*

# Line highlight

- Toutes les lignes, puis 1, puis 4, puis 6

```
1  ```{r}
2  #| code-line-numbers: "|1|4|6"
3
4  <Programme>
5  ```
```

```
1  data("mtcars")
2  x <- mtcars$wt
3  y <- mtcars$mpg
4  plot(x, y)
5  z <- mtcars$disp
6  plot(x, z)
```

# Code animation

- En option du titre des slides avec un code animé: `{auto-animate="true"}:`
  - `## Titre slides {auto-animate="true"}`
- Pour un bon rendu, les différentes codes doivent être sur une même ligne dans chaque slide:
- Dans l'exemple qui suit, le code est toujours en dessous de *Bloc 1*, *Blocs 1 + 2*, et *Blocs 1 + 2 + 3*. Donc sur la deuxième ligne de la slide.

# Code Animations

- Bloc 1

```
1 data("mtcars")
```



# Code Animations

- Blocs 1 + 2

```
1 data("mtcars")  
2 x <- mtcars$wt  
3 y <- mtcars$mpg
```

# Code Animations

- Blocs 1 + 2 + 3

```
1 data("mtcars")
2 x <- mtcars$wt
3 y <- mtcars$mpg
4 plot(x, y)
```

# Code Animations

- On peut ajouter des commentaires (**code annotation**): [\[Lien\]](#).
- Pour l'ensemble du programme (blocs 1 + 2 + 3):

```
152  ````{r}
153  #| eval: false
154
155  data("mtcars")      #<1>
156  x <- mtcars$wt      #<2>
157  y <- mtcars$mpg    #<2>
158  plot(x, y)         #<3>
159  ````
160
161  1. On ouvre la base
162  2. On définit les coordonnées
163  3. On exécute le scatter plot
164
165
```

# Code Animations

- On peut ajouter des commentaires [[Lien](#)]

```
1 data("mtcars")
```

①

① On ouvre la base

# Code Animations

- On peut ajouter des commentaires [[Lien](#)]

```
1 data("mtcars")  
2 x <- mtcars$wt  
3 y <- mtcars$mpg
```

①

②

- ① On ouvre la base
- ② On définit les coordonnées

# Code Animations

- On peut ajouter des commentaires avec [\[Lien\]](#)

```
1 data("mtcars")  
2 x <- mtcars$wt  
3 y <- mtcars$mpg  
4 plot(x, y)
```

①

②

③

- ① On ouvre la base
- ② On définit les coordonnées
- ③ On génère un scatter plot

# Elements d'affichage

# Insertion du contenu d'un lien

- Très bonne idée.
- Le contenu d'un lien est inséré dans la présentation mais avec la possibilité de le fermer directement (croix en haut à droite).
- Dans le yaml, pour activer cette option: **preview-links: true**.

[https://mthevenin.gitlab.io/support\\_quarto/](https://mthevenin.gitlab.io/support_quarto/)



# Affichage incrémental des éléments d'une liste (I)

- yml: option `incremental: true`
- Classe css avant la liste: `::: {.incremental}` ou `::: {.nonincremental}`
- Premier élément de la liste
- Second élément de la list

# Affichage incrémental des éléments d'une liste (II)

- On peut ajouter des effets avec la classe `{ .fragment <option> }` à laquelle on ajoute une classe d'effet, par exemple `{ .fragment .fade-left }`
- Exemples:
- `{ .fragment .fade-up }`
- `{ .fragment .fade-left }`

# Réduire la taille d'affichage d'une slide

- Dans le titre de la slide, on utilise la classe css `{.smaller}`
- Ici on a écrit:
  - `## Réduire la taille d'affichage d'une slide {.smaller}`

# Dépasser la limite verticale des slides de la présentation

- On peut contourner la limite verticale de la slide en la faisant défiler vers le bas.
- Dans le yaml de la présentation, on utilise l'option `scrollable: true`

```
1 format:  
2   revealjs:  
3     scrollable: true
```

- Bavardage 1
- Bavardage 2
- Bavardage 3
- Bavardage 4
- Bavardage 5
- Bavardage 6
- Bavardage 7
- Bavardage 8
- Bavardage 9

# Faciliter la mise en page dans une slide

Important car le périmètre affichage limité (déjà quelques pistes dans les points précédents avec `{.smaller}` ou `scrollable`)

# Onglet

- [Lien support principal.](#)
- Onglet: `::: {.panel-tabset}`.

Programme

Output

```
1 ```{r}
2 #| eval: false
3
4
5 data("mtcars")
6 x <- mtcars$wt
7 y <- mtcars$mpg
8 plot(x, y)
9 ```
```

# Multicolonne

- Affichage en plusieurs colonnes `::: {.columns}` puis `::: {.column}`.
- A réserver à du texte.
- [Lien support](#)

## Liste 1

- Item1
- Item2
- Item3

## Liste 2

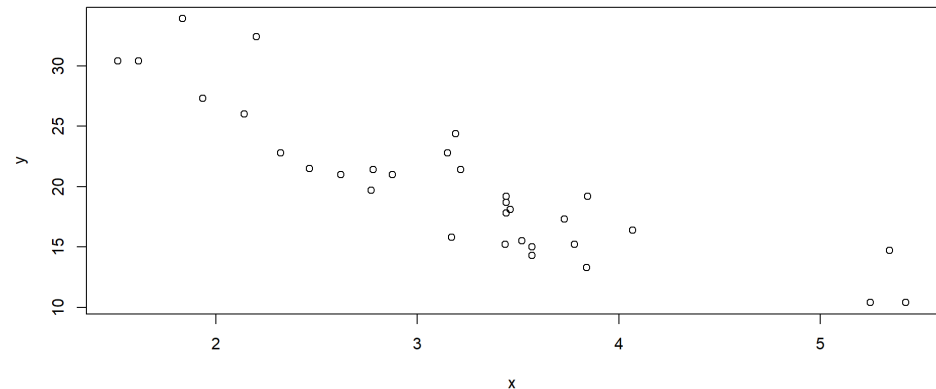
- Item1
- Item2
- Item3



# Option localisation du report de l'output

- Pour un graphique, un extrait de base, .... : option `#| output-location: option`
  - Options: `column` (à droite), `fragment` (en dessous avec touche ->), `column-fragment` (à droite avec touche ->), `slide` (slide suivante).
- Ici avec l'option `#| output-location: column-fragment`.

```
1 data("mtcars")
2 x <- mtcars$wt
3 y <- mtcars$mpg
4 plot(x, y)
```



# Theming

# Les thèmes revealjs

- 11 thèmes sont disponibles, dont un thème par défaut [\[Lien\]](#).
- On peut changer de thèmes avec l'option **theme:**  
**nom\_theme**.
- Ici le thème est **dark**:

```
1 ---
2 format:
3   revealjs:
4     theme: dark
5 ---
```

- On peut modifier le thème choisit avec un fichier `.css` ou `.scss` [\[Lien\]](#)

# Changement du background d'une slide

- On peut également changer la couleur de fond d'une ou plusieurs slide: Au titre de la slide ajouter: `{background-color=couleur}`<sup>1</sup>
  - Ici:
    - `## Changement du background d'une slide {background-color="darkturquoise"}`
    - `## Changement du background d'une slide {background-color=#00CED1}`

1. un code hexadécimal ou une couleur css prédéfinie: "nom\_couleur" [lien: liste nom couleur css dans couleur nommée]

# Divers

- A ajouter des points ultérieurement, au cas où

# Plusieurs options dans le titre de la slide

- Ajouter les options dans une même accolade:
  - Exemple pour la slide sur la couleur de fond: `{background-color}` et `{.smaller}` => `{background-color=00CED1 .smaller}`

```
1 # Changement du background d'une slide: {background-color=00CED1 .smaller}
```

# Numéro des slides

- Dans le yaml, ajouter l'option `slide-number:true`