

Outil de génération de graphiques de l'IET – Documentation

Dominic Rivest

6 mai 2025

Table des matières

1	Fonctionnement général	2
1.1	Fichiers d'entrée	2
1.2	Modification des paramètres généraux	3
2	Particularité des types de graphique	6
3	Téléchargement en lot des figures affichées	9
4	Publication des figures dans <i>Chart-Studio</i>	9

1 Fonctionnement général

L'outil de génération de graphiques de l'IET fonctionne sur un serveur hébergé par render.com. L'outil est codé dans le langage python principalement à l'aide des librairies montréalaises Dash et Plotly. Il est hébergé à l'adresse <https://generateur-de-graphiques-iet.onrender.com/>.

Le code est hébergé sur GitHub à l'adresse <https://github.com/domrivest/genGraphIET>.

1.1 Fichiers d'entrée

L'outil prend en entrée des fichiers texte séparés en deux portions : les métadonnées et les données. Les métadonnées définissent le titre, le type, les axes, les titres d'axe, la source et certaines options d'affichage des figures. Voici un exemple de fichier de définition d'un graphique à bandes horizontal :

```
metadata;chart.title;Industrial energy use by industry (2000, 2010 and 2020)
metadata;chart.subtitle
metadata;chart.xLabel;PJ
metadata;chart.yLabel
metadata;chart.source;OEE 2023
metadata;chart.section
metadata;chart.type;bar.grouped.horizontal
metadata;chart.level_1;year
metadata;chart.level_2;sector
metadata;chart.note;Percentages shown in the vertical axis represent the share of
    total energy use in the sector (total is not 100% due to rounding)
metadata;data.source
metadata;data.file
data;sector;year;value
data;Mines, carrières, extraction pétrole et gaz (40.9%);2000;505.790011
data;Mines, carrières, extraction pétrole et gaz (40.9%);2010;974.032479
data;Mines, carrières, extraction pétrole et gaz (40.9%);2020;1439.04142
data;Pâtes et papiers (14.1%);2000;867.656198
data;Pâtes et papiers (14.1%);2010;552.674536
data;Pâtes et papiers (14.1%);2020;511.646516
data;Autres manufacturiers (12.1%);2000;563.601817
data;Autres manufacturiers (12.1%);2010;472.032845
data;Autres manufacturiers (12.1%);2020;427.320113
```

```

data;Raffinage du pétrole (7.7%);2000;342.61854
data;Raffinage du pétrole (7.7%);2010;350.948124
data;Raffinage du pétrole (7.7%);2020;272.4289
data;Fonte et affinage (7.6%);2000;231.263348
data;Fonte et affinage (7.6%);2010;238.389441
data;Fonte et affinage (7.6%);2020;268.126911
data;Produits chimiques (7.0%);2000;260.302
data;Produits chimiques (7.0%);2010;248.356303
data;Produits chimiques (7.0%);2020;245.977563
data;Fer et acier (15.1%);2000;260.134
data;Fer et acier (15.1%);2010;213.113026
data;Fer et acier (15.1%);2020;179.423453
data;Construction (2.9%);2000;51.320932
data;Construction (2.9%);2010;73.443483
data;Construction (2.9%);2020;102.305474
data;Ciment (1.4%);2000;67.066
data;Ciment (1.4%);2010;57.34161
data;Ciment (1.4%);2020;50.731356
data;Foresterie (0.6%);2000;17.170664
data;Foresterie (0.6%);2010;22.318411
data;Foresterie (0.6%);2020;21.228021

```

1.2 Modification des paramètres généraux

La définition des variables et de leurs étiquettes anglaises et françaises se fait dans le fichier *colors.csv*. Il est situé dans le dossier *assets* du répertoire GitHub. Outre cela, il existe deux types de paramètres modifiables par l'utilisateur du générateur de graphiques : les paramètres modifiés dans l'application et les paramètres modifiés dans les fichiers texte directement.

Le premier type inclut les paramètres suivants :

- Produire les figures en français (Booléen) : Utilise la colonne *label_fr* du fichier *colors.csv* si vrai.
- Afficher la source lorsqu'elle est renseignée dans le fichier (Booléen) : Affiche en bas de la figure la source en taille 10 si *metadata;data.source* est renseigné.
- Afficher le titre lorsqu'il est renseigné dans le fichier (Booléen) : Affiche au-dessus de la figure

le titre si *metadata;chart.title* est renseigné.

- Taille de police (Choix d'un entier) : Contrôle la taille des éléments texte de manière relative (ex : le titre sera toujours plus gros que les titres d'axe).
- Spécifier la hauteur, la largeur et la résolution des figures produites (entrées d'entiers positifs). La résolution s'applique lors du téléchargement en PNG et est un simple multiplicateur des dimensions de la figure qui ne modifie pas l'échelle des éléments. Ainsi une figure de 1000x400 pixels avec une résolution de 10 deviendra une image de 10000x4000 pixels avec la même apparence.

Bien que les paramètres par défaut font bien le travail en général, certains paramètres peuvent être modifiés en spécifiant des valeurs dans les fichiers textes propres aux figures. Il s'agit d'ajouter une ligne contenant le paramètre à modifier avec la bonne syntaxe. Des exemples sont donnés ci-dessous.

Modifications des axes :

- Type d'axe (*metadata;xaxes.type* et *metadata;yaxes.type*) ; Valeurs possibles : ("-" | "linear" | "log" | "date" | "category" | "multicategory") ; Défaut : "-" (S'ajuste automatiquement) ; Documentation : Documentation Plotly ; Exemple : *metadata;xaxes.type;category*
- Mode de graduation (*metadata;xaxes.tickmode* et *metadata;yaxes.tickmode*) ; Valeurs possibles : ("auto" | "linear" | "array") ; Défaut : "auto" ; Documentation : Documentation Plotly ; Exemple : *metadata;xaxes.tickmode;linear*
- Valeur initiale de graduation (*metadata;xaxes.tick0* et *metadata;yaxes.tick0*) ; Valeurs possibles : Nombre ou catégorie ; À utiliser avec *dtick*. À noter que Plotly ne laissera pas de trou dans l'axe, ainsi, si la valeur initiale commence après la donnée initiale et que l'intervalle entre les graduations est trop petit, il y aura tout de même des graduations en-deça de la valeur initiale. ; Documentation : Documentation Plotly ; Un exemple est montré ci-bas.
- Delta entre les graduations (*metadata;xaxes.dtick* et *metadata;yaxes.dtick*) ; Valeurs possibles : Nombre positif ; À utiliser avec *tick0* ; Voir la documentation pour les axes de type "log" : Documentation Plotly ; Un exemple est montré ci-bas.
- Nombre maximal de graduations sur l'axe (peut être inférieur au nombre spécifié) (*metadata;xaxes.nticks* et *metadata;yaxes.nticks*) ; Valeurs possibles : Entiers positifs ; Documentation : Documentation Plotly ; Un exemple est montré ci-bas.
- Valeurs de graduation spécifiques (*metadata;xaxes.tickvals* et *metadata;yaxes.tickvals*) ; Valeurs possibles : Liste de nombres ou éléments de texte séparés par une virgule ; Documenta-

tion : Documentation Plotly ; Un exemple est montré ci-bas.

- Rotation des étiquettes d'axe (*xaxes.tickangle* et *yaxes.tickangle*) ; Valeurs possibles : Nombre ; Change l'angle des étiquettes par rapport à l'horizontal ; Documentation : Documentation Plotly ; Exemple : *metadata;xaxes.tickangle;90*
- Mode d'ajustement initial de l'amplitude des axes (*xaxes.rangemode* et *yaxes.rangemode*) ; Valeurs possibles : ("normal" | "tozero" | "nonnegative") ; Si "normal", l'échelle est affichée en fonction des données du graphique. Si "tozero", l'échelle débute à 0. Si non négatif, l'axe affichera uniquement des valeurs positives. ; Documentation : Documentation Plotly ; Exemple : *metadata;xaxes.rangemode;tozero*
- Valeurs limites des axes (*xaxes.range* et *yaxes.range*) ; Valeurs possibles : Nombre, Nombre ; Fixe les valeurs minimales et maximales de l'axe. ; Documentation : Documentation Plotly ; Exemple : *metadata;xaxes.range;0,0.85*

Exemple d'ajustement de la graduation d'un axe des y

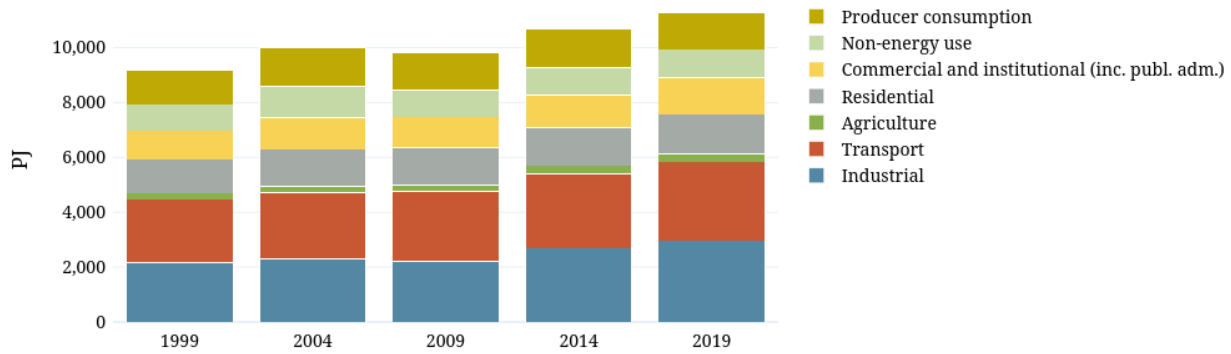
Voici le code ajouté au fichier texte du graphique pour spécifier la graduation par valeur initiale et par delta :

```
metadata;yaxes.tickmode;linear
metadata;yaxes.tick0;500
metadata;yaxes.dtick;2500
```

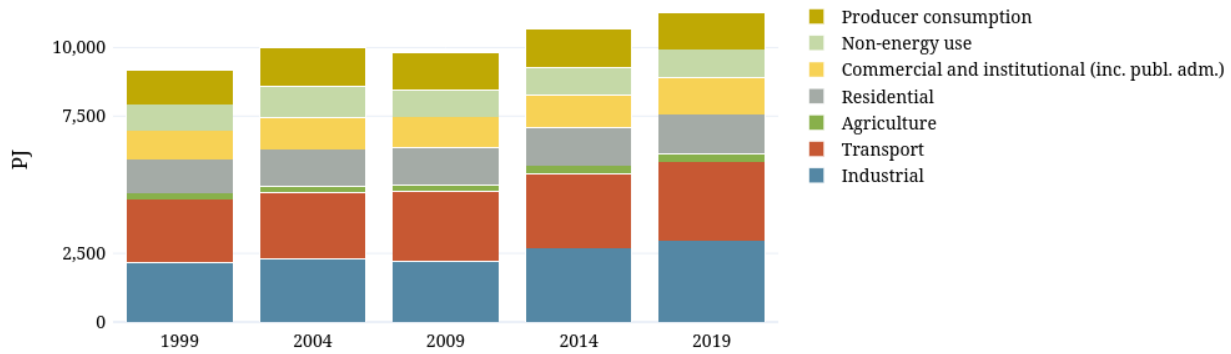
Voici le code ajouté au fichier texte du graphique pour spécifier la graduation pour des valeurs spécifiques :

```
metadata;yaxes.type;linear
metadata;yaxes.tickvals;0,2500,7500,10000
```

La figure 1 montre la différence entre les deux méthodes.



(a) Valeurs de graduation spécifiées par valeur initiale et par delta



(b) Valeurs de graduation spécifiques

FIGURE 1 – Exemple d'application des paramètres d'axe sur l'axe des y

2 Particularité des types de graphique

Les classes de graphique (*metadata;chart.type*) présentement supportées sont les suivantes :

- Graphique en aires – *area*
- Graphique à bandes groupées horizontal – *bar.grouped.horizontal*
- Graphique à bandes superposées horizontal – *bar.stacked.horizontal*
- Graphique à bandes superposées – *bar.stacked*
- Graphique à bandes groupées – *bar.grouped*
- Graphique à bandes superposées et groupées – *bar.grouped.stacked*

- Graphique à lignes – *line*
- Nuage de points – *scatter*
- Graphique en pointe de tarte – *pie*

Spécifications par type de graphique :

Graphique en aires :

Voici un exemple des deux premières lignes de données :

```
data;year;Natural gas;Crude oil;Primary electricity;Coal;Gas plant natural gas
liquids (NGL's)
data;2001;3381.3;4170.052;1364.567;1314.497;491.213
```

Graphique à bandes groupées horizontal et graphique à bandes superposées horizontal

Ici, la langue des variables doit être différenciée dans les fichiers texte. Voici un exemple des sept premières lignes de données :

```
data;usage;year;value
data;Chauffage (56.6%);2000;603.1
data;Chauffage (56.6%);2010;528.9
data;Chauffage (56.6%);2020;687.8
data;Matériel auxiliaire (15.1%);2000;87.3
data;Matériel auxiliaire (15.1%);2010;138.1
data;Matériel auxiliaire (15.1%);2020;183.8
```

Graphique à bandes superposées

Voici un exemple des trois premières lignes de données :

```
data;service;Electricity;Natural gas;Heating Oil;Wood;Other (including coal and
propane)
data;Space heating;260;544;52;165;14
data;Water heating;71;197;7;5;1
```

Graphique à bandes groupées

Voici un exemple des trois premières lignes de données :

```
data;province;year;value
```

```
data;BC;1990;15.73
```

```
data;BC;2005;15.01
```

Graphique à bandes superposées et groupées

Voici un exemple des quatre premières lignes de données :

```
data;province;year;1990 (energy);1990 (non-energy);2005 (energy);2005 (non-energy)
;2021 (energy);2021 (non-energy)
data;BC;1990;41654;10129;0;0;0;0
data;BC;2005;0;0;50013;12960;0;0
data;BC;2021;0;0;0;0;51814;7622
```

Graphique à lignes et nuage de points

Les deux types de figures se définissent de la même manière, mais le nuage de points n'a pas de lignes reliant les points. Voici un exemple des deux premières lignes de données :

```
data;reference;month;Mt
data;Western Canadian Select (WCS);2014-01-01;65.69
```

Graphique en pointe de tarte

Pour les figures en pointe de tarte, le pourcentage par pointe est affiché par défaut. Il est cependant possible d'afficher les valeurs directement ou les deux simultanément. Cependant, cette dernière option prend plus de place dans la figure.

Voici les premières lignes de données pour le graphique par défaut :

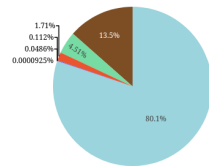
```
data;Gaz;valeur;valeur %
data;Gas_CO2;537174;80,1%
data;Gas_CH4;90510;13,5%
```

Pour afficher la valeur uniquement (et le pourcentage dans le hover), on utilise :

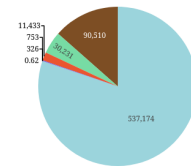
```
metadata;chart.textinfo;value;
metadata;chart.hoverinfo;label+percent+value;
data;Gaz;valeur;valeur %
data;Gas_CO2;537174;80,1%
```


Pour afficher la valeur et le pourcentage, on utilise :

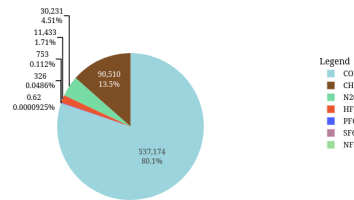
```
metadata;chart.textinfo;value+percent;
metadata;chart.hoverinfo;label+percent+value;
data;Gaz;valeur;valeur %
data;Gas_CO2;537174;80,1%
```



(a) Par défaut



(b) Avec valeur uniquement



(c) Valeur et pourcentage

FIGURE 2 – Configuration des figures en pointe de tarte.

3 Téléchargement en lot des figures affichées

Dans l'outil de génération, le téléchargement en lot d'un maximum de 100 figures affichés est possible en format vectoriel PDF ou SVG. Dû aux limitations matérielles du serveur, à l'efficacité actuelle de la librairie d'export en PNG et à la résolution d'export souhaitée, cette option n'est pour l'instant pas disponible.

4 Publication des figures dans *Chart-Studio*

Chart-Studio est une plateforme développée par *Plotly* pour héberger et modifier des figures en ligne. Cette plateforme est la base de la stratégie d'intégration des figures du générateur dans Wordpress. Elle permet de conserver une copie d'un graphique sur le web et à des ressources externes comme Wordpress ou quelconque site internet d'y accéder. Le bouton d'export vers Chart-Studio envoie

tous les figures affichées vers un compte Chart-Studio avec leur nom de fichier et un préfixe, si la case est remplie.

Une fois les figures exportées, on peut les retrouver sur Chart-Studio et utiliser la fonction exporter (*share*) pour retrouver le code HTML ou sa version iFrame qui sera à intégrer dans la page Wordpress. Voici à quoi le code et la plateforme ressemblent lorsqu'on veut partager un graphique pour l'imbriquer :

HTML brut :

```
<div>
<a href="https://plotly.com/~domrivest/186/?share_key=MPPrctkTcDTUV6joL7Ek2W" target
  ="_blank" title="testPiefig5.2" style="display: block; text-align: center;"></a>
<script data-plotly="domrivest:186" sharekey-plotly="MPPrctkTcDTUV6joL7Ek2W" src="
  https://plotly.com/embed.js" async></script>
</div>
```

iFrame :

```
<iframe width="900" height="800" frameborder="0" scrolling="no" src="//plotly.com/~
  domrivest/186.embed"></iframe>
```

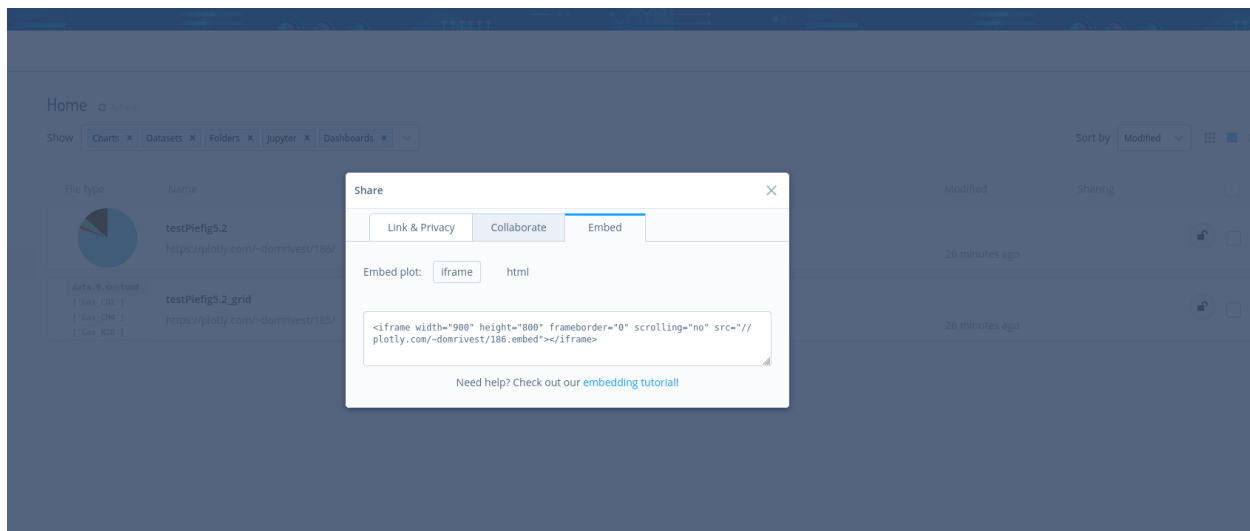


FIGURE 3 – Interface de chart-studio

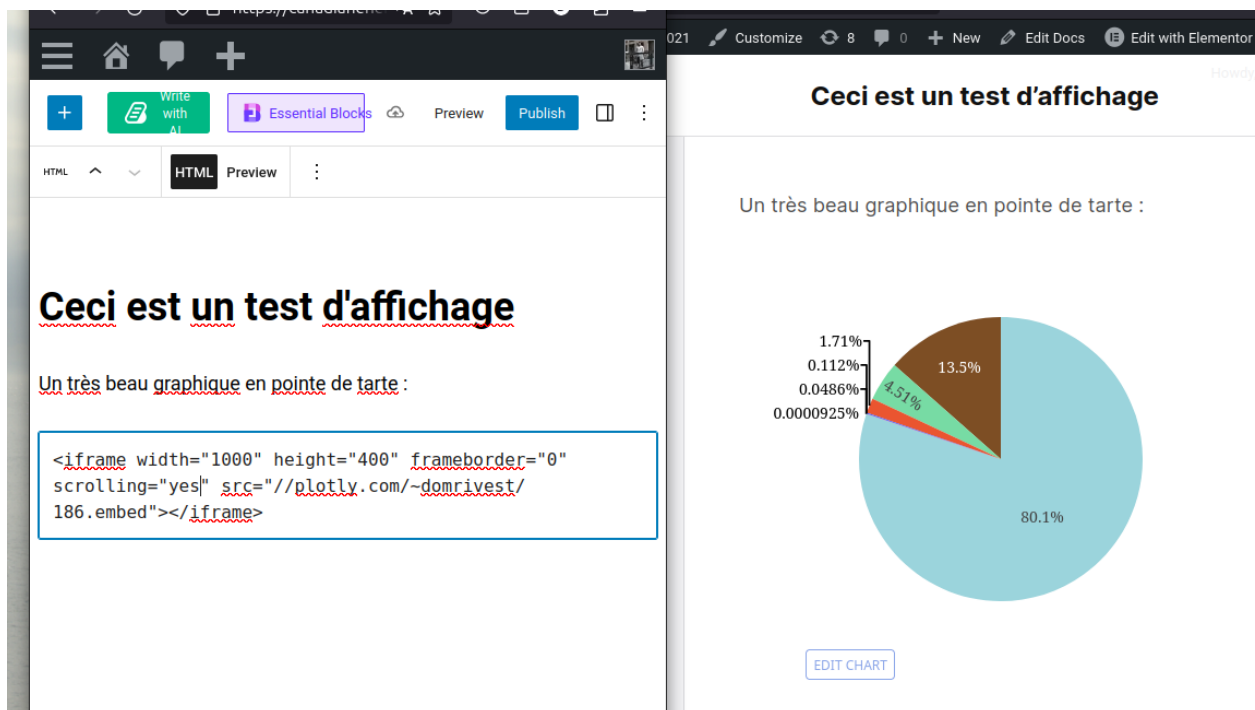


FIGURE 4 – Intégration dans Wordpress