Diagrammes univariés avec Seaborn

Maîtriser Seaborn : Démystifier les diagrammes complexes · https://moncoachdata.com/tutos/diagrammes-univaries-avec-seaborn/

Les **diagrammes univariés avec Seaborn** servent à visualiser et analyser une seule variable à la fois. Ils permettent de comprendre sa distribution et sa tendance centrale, d'identifier les valeurs aberrantes et de repérer des schémas au sein de la variable. Les exemples utilisent le jeu de données `tips` :

```python
import seaborn as sns
import matplotlib.pyplot as plt

tips = sns.load_dataset("tips")
```

Python

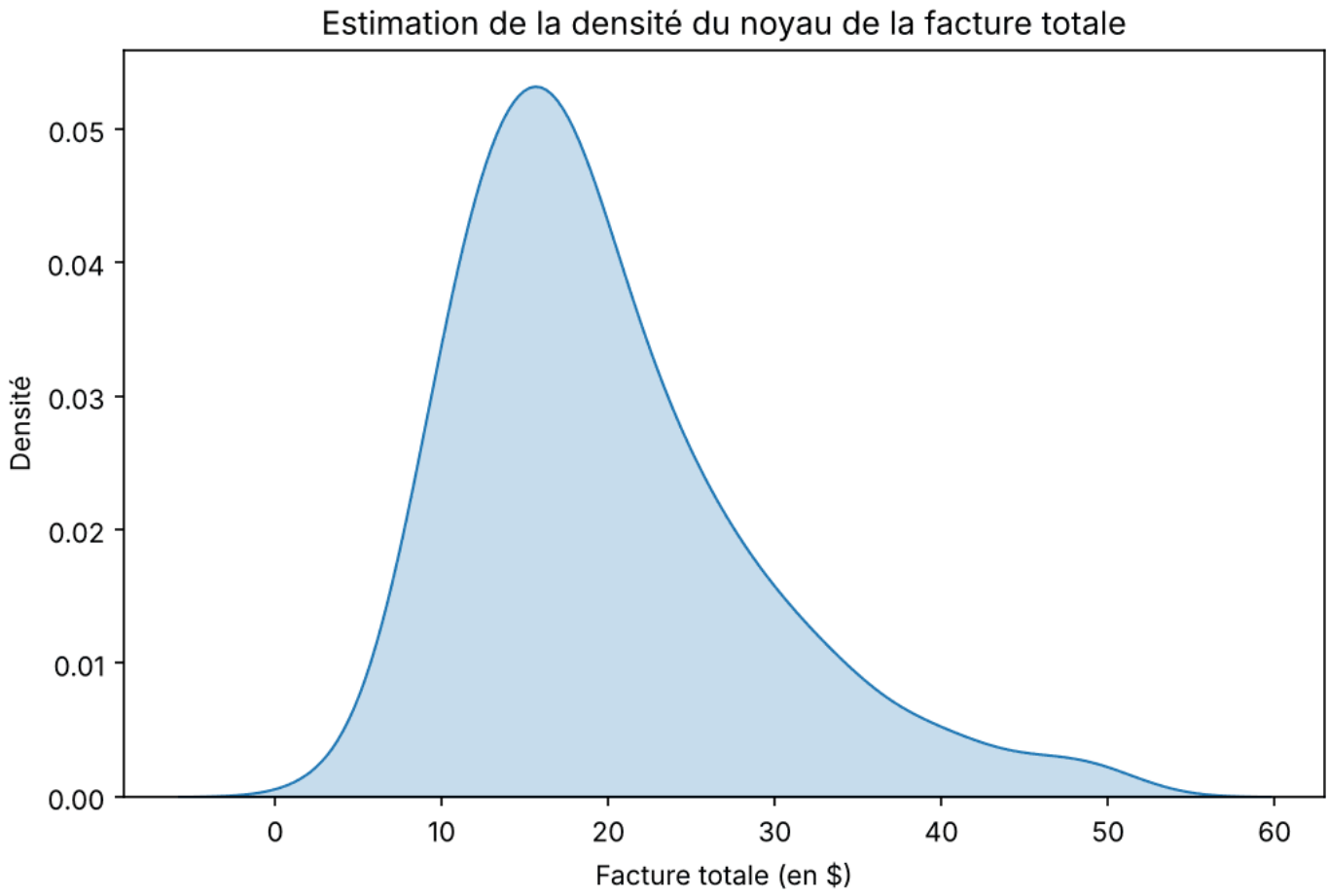
KDE Plot

Un diagramme KDE (*Kernel Density Estimation*) estime la fonction de densité de probabilité d'une variable continue. La variable est sur l'axe x, et l'axe y représente la densité de probabilité estimée. L'aire totale sous la courbe vaut 1, ce qui signifie que la distribution est normalisée.

Utilise `sns.kdeplot(data=dataframe[colonne])` pour créer un KDE plot. Tu peux le personnaliser avec des paramètres comme la largeur de bande (`bw_adjust`), le noyau ou la couleur.

```python
plt.figure(figsize=(8, 5))
sns.kdeplot(data=tips["total_bill"], fill=True)
plt.title("Estimation de la densité du noyau de la facture totale")
plt.xlabel("Facture totale (en $)")
plt.ylabel("Densité")
plt.show()
```

Python



Un KDE plot donne un aperçu de la forme, des pics, des modes et de la dispersion des données. Il est précieux pour estimer la densité de probabilité des points, ce qui est utile en analyse statistique et pour les tests d'hypothèses.

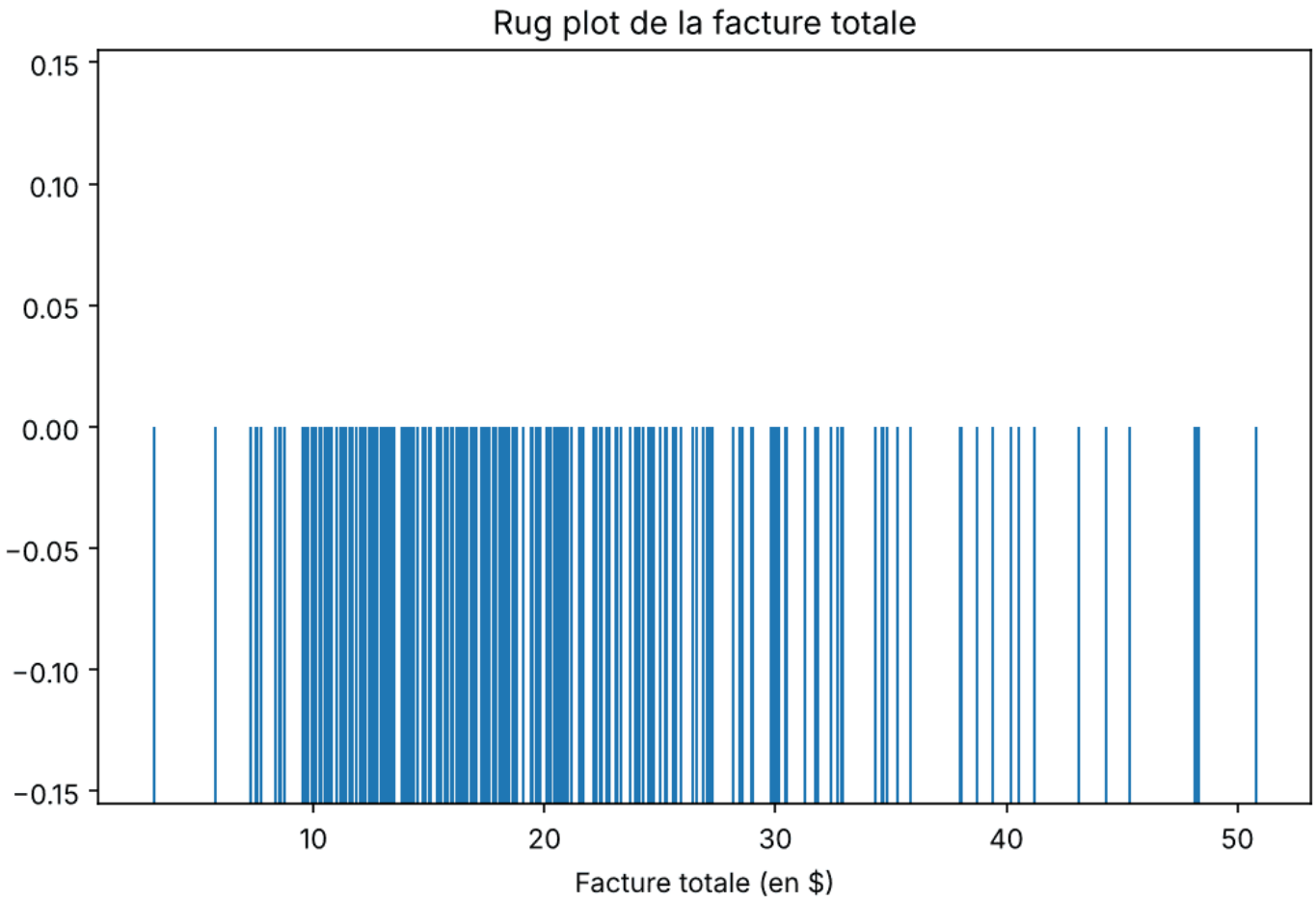
Rug Plot

Un rug plot est composé de lignes verticales (des « tics ») positionnées le long d'un axe, généralement l'axe x. Chaque tic marque l'emplacement d'un point de données, et leur proximité représente la densité : plus les tics sont serrés, plus la zone est dense.

Utilise `sns.rugplot(x=dataframe[colonne])`. Tu peux personnaliser la hauteur, la couleur et l'orientation des tics.

```python
plt.figure(figsize=(8, 5))
sns.rugplot(x=tips["total_bill"], height=0.5)
plt.title("Rug plot de la facture totale")
plt.xlabel("Facture totale (en $)")
plt.show()
```

Python



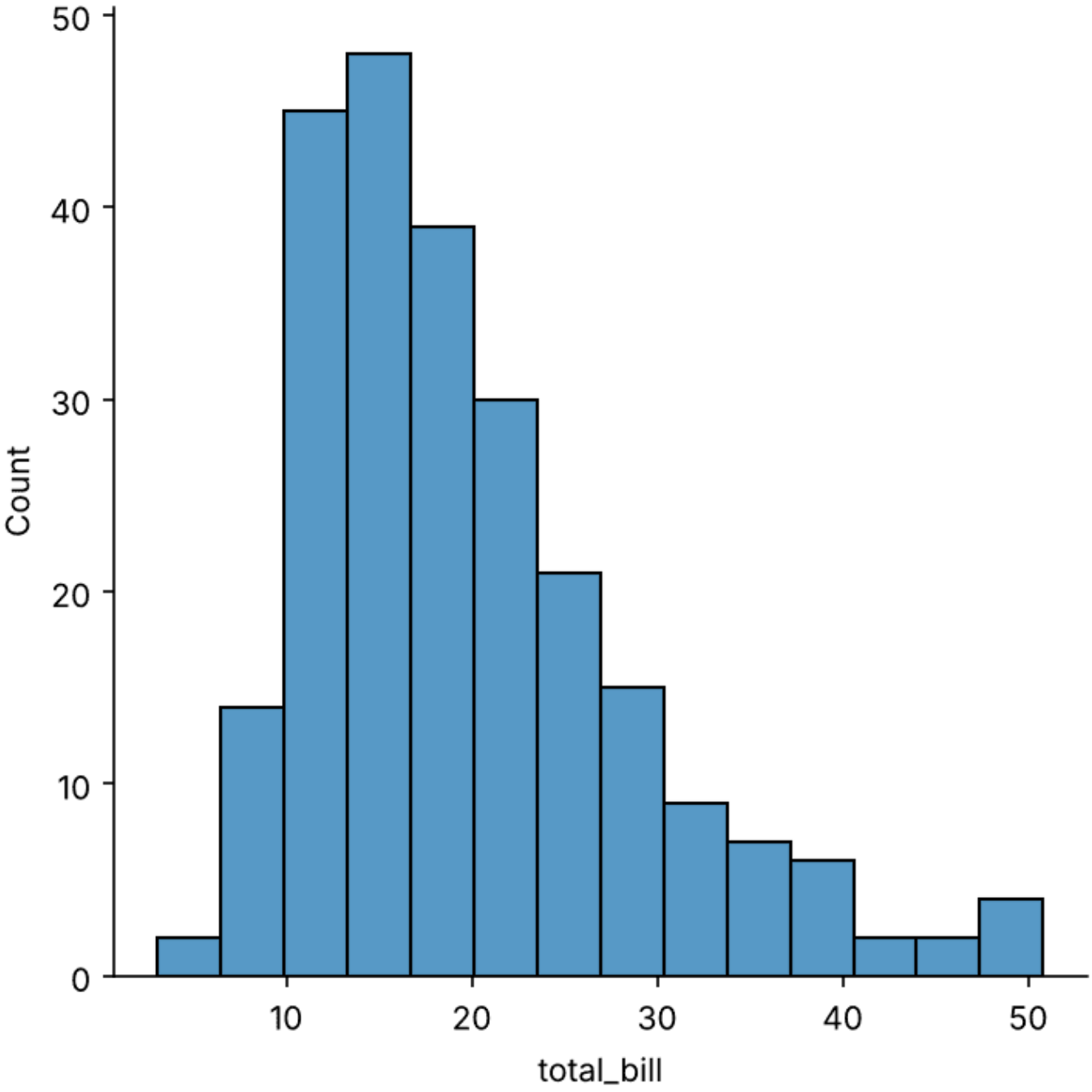
Le KDE plot montre un pic autour de 15 à 16 $. Le rug plot confirme que les factures se concentrent entre 10 et 20 $ environ, et montre la position exacte des points, y compris les rares factures élevées à droite. Il est souvent combiné avec un histogramme, un KDE plot ou un diagramme en boîte pour ajouter du contexte, et aide à repérer les valeurs aberrantes ou les zones denses.

Distribution plot (displot)

Un diagramme de distribution ressemble à un histogramme : il divise la plage de la variable numérique en intervalles (*bins*) et affiche la fréquence des points de chaque intervalle. Dans Seaborn, la fonction à utiliser est `sns.displot()`. L'ancienne fonction `distplot()` est dépréciée et supprimée.

```python
sns.displot(data=tips, x="total_bill")
plt.show()
```

Python



- Par défaut, `displot()` renvoie un histogramme. Ajoute `kde=True` pour superposer une courbe de densité, ou `kind='kde'` pour obtenir uniquement le KDE ;
- Ajoute `rug=True` pour afficher un rug plot le long de l'axe x et voir les points de données individuels ;
- Comme `displot()` est une fonction « figure-level » qui crée sa propre figure, n'appelle pas `plt.figure()` avant : utilise plutôt son paramètre `height` pour la taille, et `g.set_axis_labels()` ou `g.set(...)` pour les titres.

```python
g = sns.displot(data=tips, x="total_bill", kde=True, rug=True, height=5, aspect=1.5)
g.set_axis_labels("Facture totale (en $)", "Fréquence")
g.figure.suptitle("Répartition de la facture totale", y=1.02)
plt.show()
```

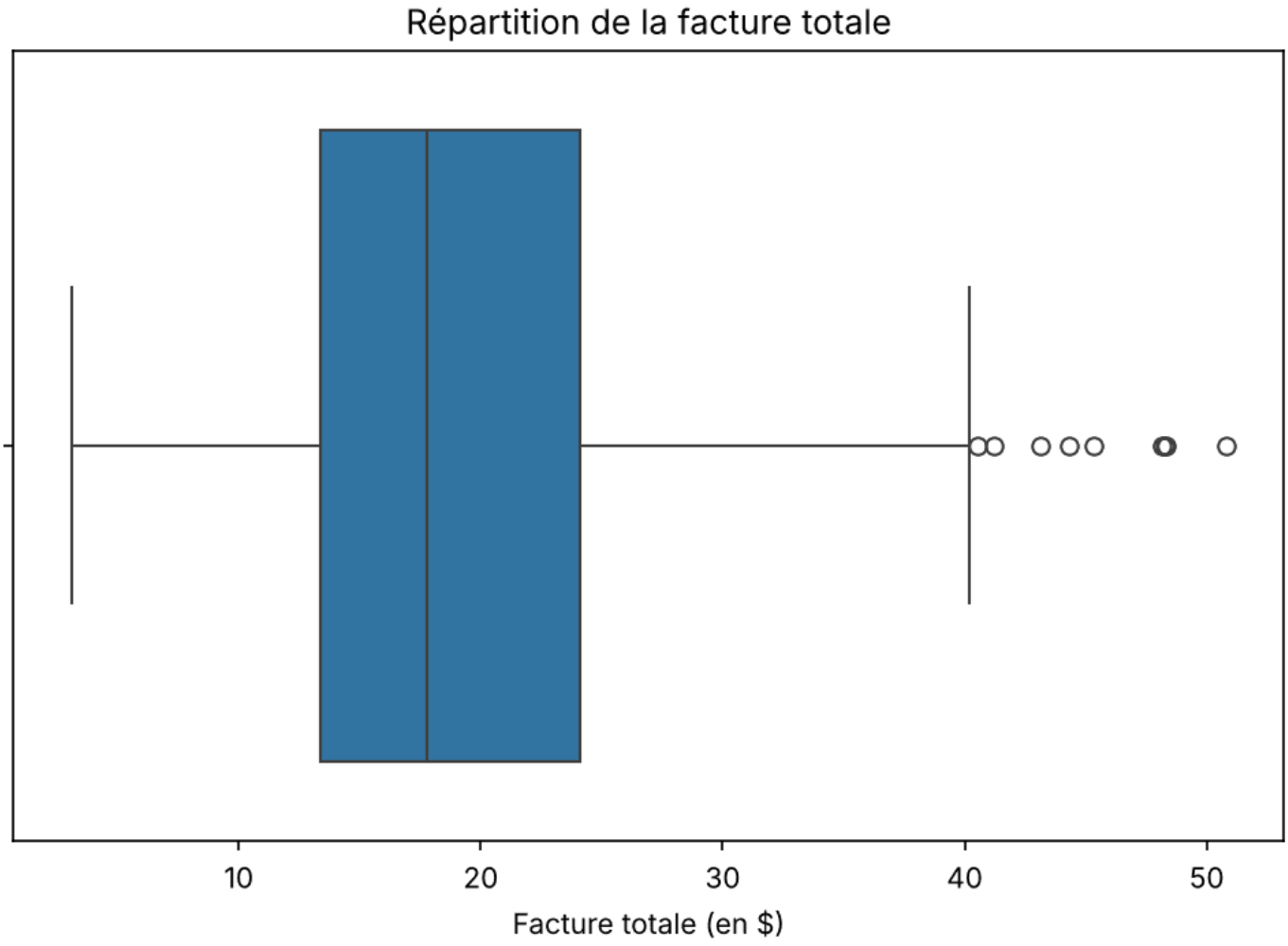
Python

Box plot et Violin plot

Ces deux diagrammes sont identiques à ceux de la section sur les diagrammes catégoriels. La seule différence : tu ne passes qu'une seule variable pour une analyse univariée.

```python
plt.figure(figsize=(8, 5))
sns.boxplot(x=tips["total_bill"])
plt.title("Répartition de la facture totale")
plt.xlabel("Facture totale (en $)")
plt.show()
```

Python



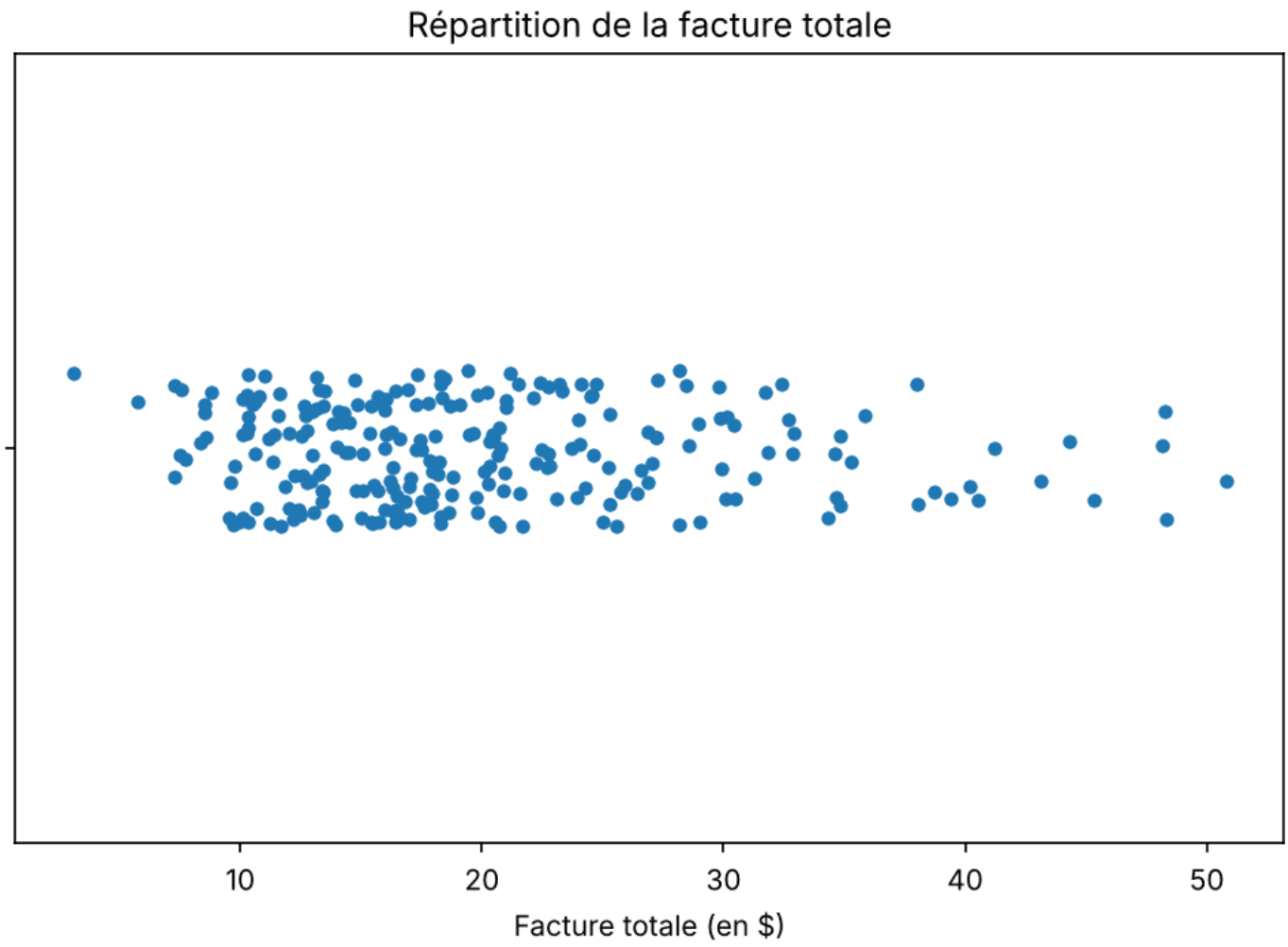
Strip Plot

Un strip plot ressemble à un swarm plot : il affiche les points de données individuels le long d'un seul axe. Mais dans un strip plot, les points peuvent se chevaucher, alors que le swarm plot les écarte et convient donc mieux aux petits jeux de données.

Utilise `sns.stripplot(x=dataframe[colonne])` (il fonctionne aussi pour des données bivariées). Le paramètre `jitter` ajoute un petit bruit aléatoire aux points pour limiter le chevauchement et améliorer la visibilité.

```python
plt.figure(figsize=(8, 5))
sns.stripplot(x=tips["total_bill"], jitter=True)
plt.title("Répartition de la facture totale")
plt.xlabel("Facture totale (en $)")
plt.show()
```

Python



Conclusion

Tu as vu dans cette série différents graphiques catégoriels et univariés : diagramme de comptage, swarm plot, point plot, cat plot, KDE plot, rug plot, box plot, distribution plot, violin plot et strip plot. Continue à explorer : Seaborn est un outil polyvalent, et tu peux faire beaucoup plus avec. Dans les prochains tutos, tu découvres les diagrammes bivariés, multivariés et matriciels.

Documentation : visualiser des distributions avec Seaborn.

Pour aller plus loin, consulte aussi notre article sur Seaborn pour la data science.


SEABORN