Diagrammes bivariés avec Seaborn

Maîtriser Seaborn : Démystifier les diagrammes complexes · https://moncoachdata.com/tutos/diagrammes-bivaries/

Les **diagrammes bivariés avec Seaborn** visualisent et analysent la relation entre deux variables à la fois. Ils servent à explorer comment deux variables sont liées ou corrélées. Les tracés les plus courants dans Seaborn sont `sns.scatterplot(x, y, data)` et `sns.lineplot(x, y, data)`, pour les diagrammes de dispersion et les diagrammes linéaires. Les exemples utilisent le jeu de données `tips` :

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

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

Python

```python
# Diagramme de dispersion : pourboire en fonction de la facture
sns.scatterplot(x="total_bill", y="tip", data=tips)
plt.show()

# Diagramme linéaire : facture moyenne selon la taille du groupe
sns.lineplot(x="size", y="total_bill", data=tips)
plt.show()
```

Python

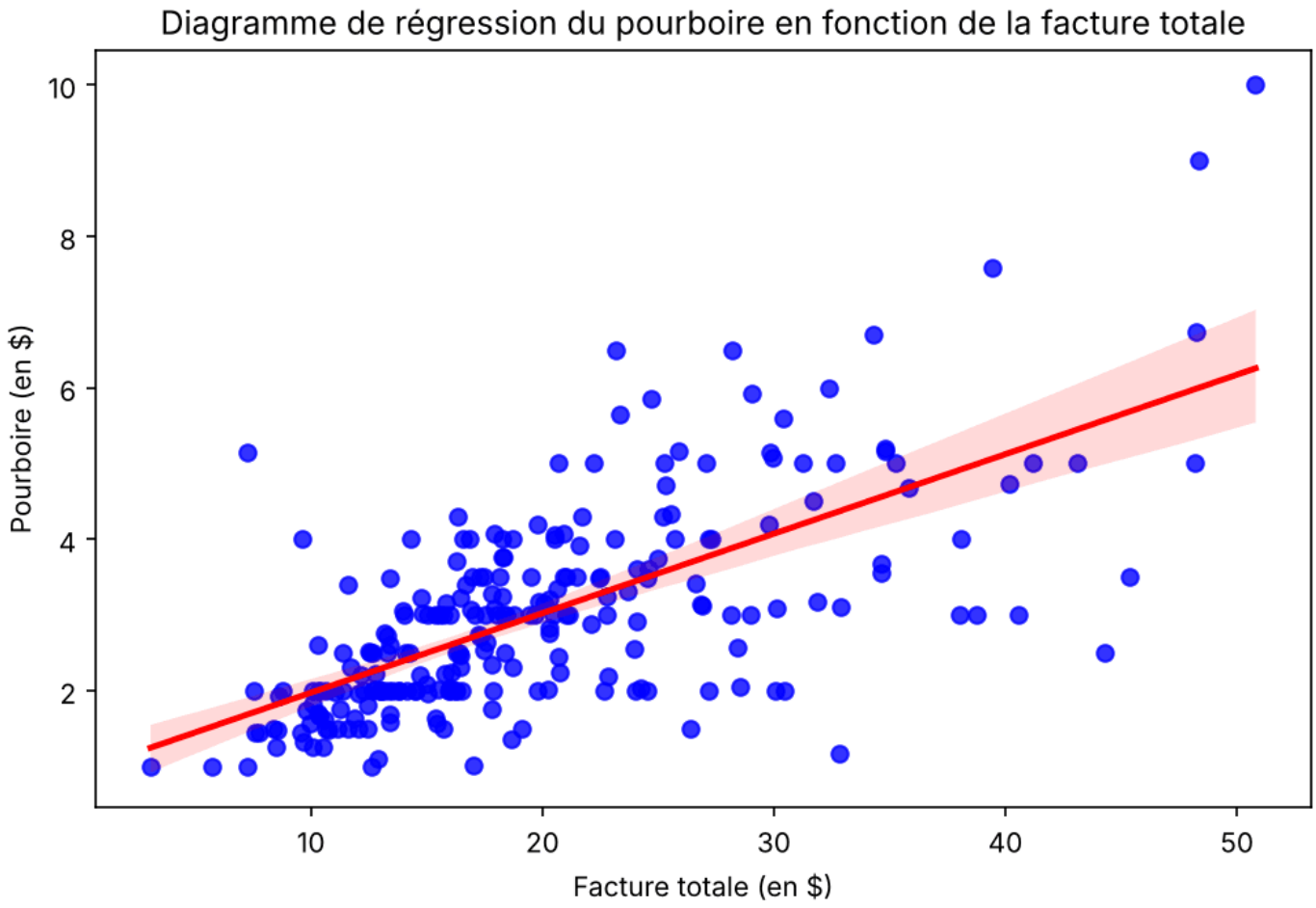
Diagramme de régression

Un diagramme de régression se concentre sur la relation entre deux variables numériques : la variable indépendante (souvent sur l'axe x) et la variable dépendante (sur l'axe y). Les points de données individuels sont affichés, et l'élément central est la droite (ou la courbe) de régression, qui représente le modèle mathématique le mieux adapté pour décrire la relation entre les variables.

Utilise `sns.regplot(x=..., y=..., data=...)` :

```python
plt.figure(figsize=(8, 5))
sns.regplot(x="total_bill", y="tip", data=tips,
            scatter_kws={"color": "blue"}, line_kws={"color": "red"})
plt.title("Diagramme de régression du pourboire en fonction de la facture totale")
plt.xlabel("Facture totale (en $)")
plt.ylabel("Pourboire (en $)")
plt.show()
```

Python



La droite de régression représente le modèle linéaire le mieux adapté pour prédire le pourboire à partir du montant de la facture. Les points montrent comment les données se regroupent autour de la droite. Ce diagramme est utile pour comprendre la relation linéaire entre deux variables.

Joint Plot

Un joint plot (diagramme conjoint) combine des diagrammes de dispersion, des histogrammes et des diagrammes de densité pour visualiser la relation entre deux variables numériques. Son élément central est un diagramme de dispersion qui affiche les deux variables l'une par rapport à

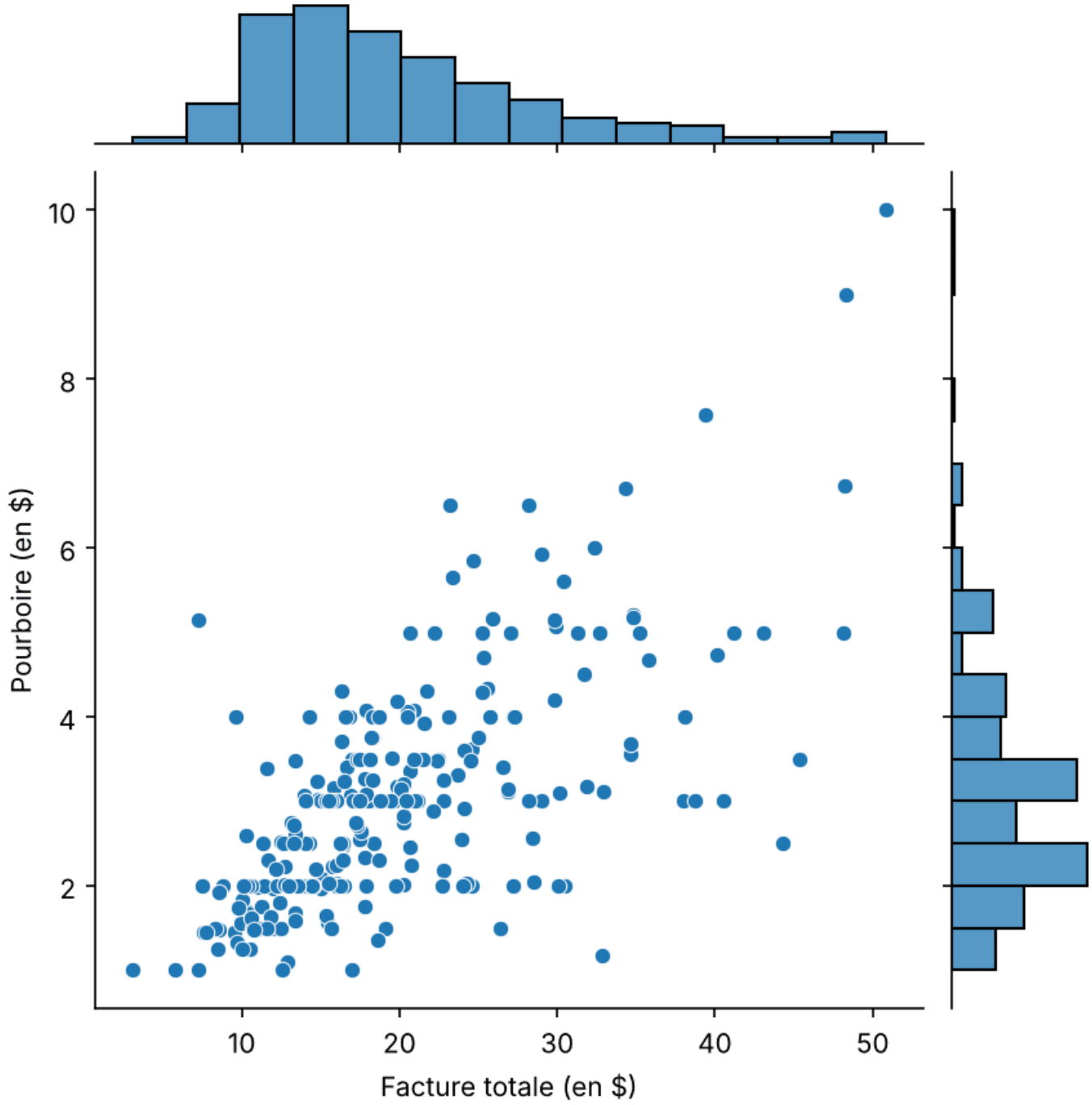
l'autre. Le long des axes x et y, des histogrammes ou des KDE montrent la distribution de chaque variable séparément (ce sont les tracés « marginaux »).

Utilise `sns.jointplot(x=..., y=..., data=dataframe, kind=...)`, où `kind` peut valoir `'scatter'`, `'hist'`, `'hex'`, `'kde'`, `'reg'` ou `'resid'`.

```python
g = sns.jointplot(x="total_bill", y="tip", data=tips, kind="scatter")

# jointplot crée sa propre figure : les étiquettes se règlent sur l'objet g
g.set_axis_labels("Facture totale (en $)", "Pourboire (en $)")
plt.show()
```

Python



Le diagramme de dispersion montre la relation entre les deux variables, tandis que les histogrammes marginaux montrent la distribution de chacune séparément. Comme `jointplot()` crée sa propre figure, n'utilise pas `plt.figure()` avant, et règle les étiquettes avec `g.set_axis_labels()`.

Hexbin Plot

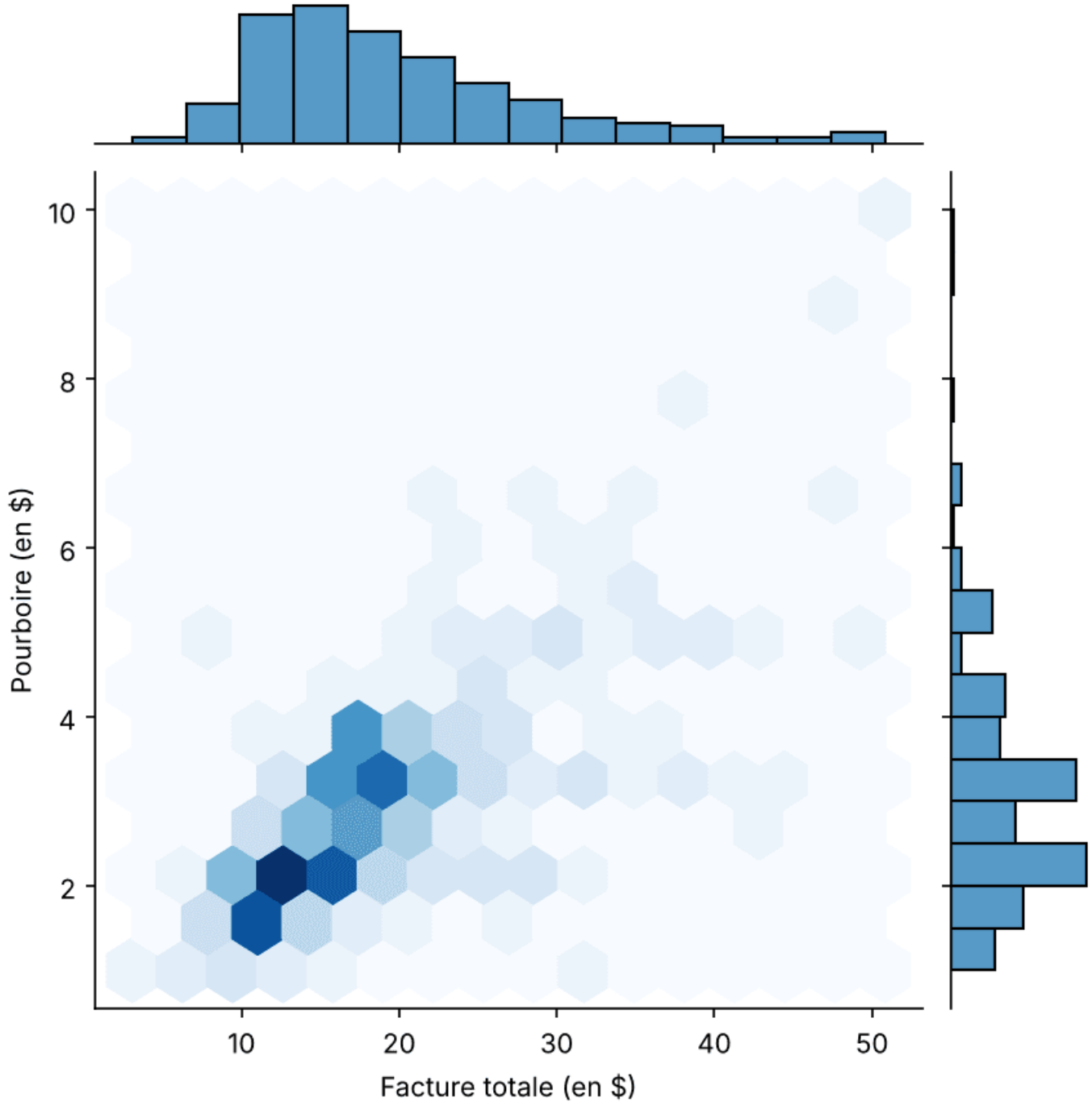
Un hexbin plot (diagramme hexagonal, pour *hexagonal binning*) regroupe les points de données dans des cases hexagonales, ce qui permet de visualiser plus efficacement leur densité. Il est particulièrement utile pour les grands jeux de données, quand un diagramme de dispersion

devient trop encombré pour être lu.

Dans Seaborn, tu le crées avec `sns.jointplot(..., kind='hex')`. Tu peux personnaliser la palette de couleurs (`cmap`), la taille de la grille (`gridsize`) et d'autres paramètres.

```python
g = sns.jointplot(x="total_bill", y="tip", kind="hex", data=tips, gridsize=15, cmap="Blues")
g.set_axis_labels("Facture totale (en $)", "Pourboire (en $)")
plt.show()
```

Python



Chaque hexagone est coloré selon la densité des points qu'il contient. Sur un grand jeu de données, ce diagramme est bien plus clair qu'un nuage de points.

La suite de la série porte sur les diagrammes multivariés avec Seaborn. Documentation : visualiser des relations statistiques avec Seaborn.

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

