Diagrammes catégoriels avec Seaborn

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

Les **diagrammes catégoriels avec Seaborn** servent à visualiser des données catégorielles, c'est-à-dire des catégories ou groupes discrets et distincts. Ce tuto passe en revue le count plot, le swarm plot, le point plot, le box plot, le violin plot et le cat plot.

Charger le jeu de données

Seaborn "aime" Pandas : contrairement à Matplotlib, tu n'as pas besoin de passer chaque série séparément (`dataframe[colonne]`). Le paramètre `data` reçoit ton `DataFrame`, et tu donnes simplement le nom des colonnes à `x` et `y`. Commençons par charger un jeu de données fourni avec Seaborn, `tips` : les pourboires laissés par les clients d'un restaurant.

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

# Charger le jeu de données "tips"
tips = sns.load_dataset("tips")
print(tips.head())
```

Python

```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

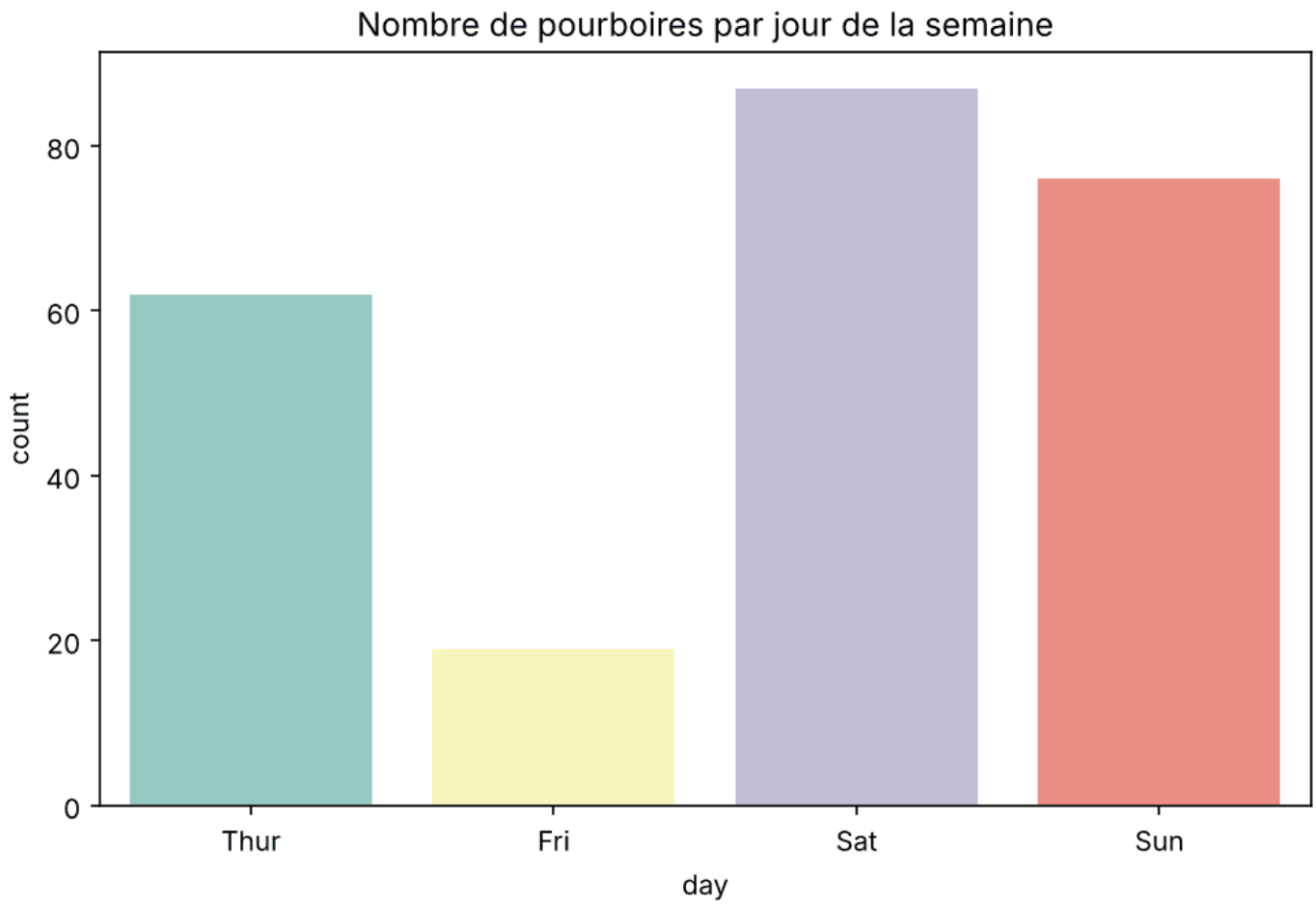
Count Plot

Le count plot (diagramme de comptage) affiche le nombre d'observations de chaque catégorie. Il est adapté pour visualiser la fréquence ou la distribution de données catégorielles. Utilise

`sns.countplot(x=..., data=dataframe)`. Voyons le nombre de lignes pour chaque jour (`day`) :

```python
plt.figure(figsize=(8, 5))
sns.countplot(x="day", data=tips, palette="Set3", hue="day", legend=False)
plt.title("Nombre de pourboires par jour de la semaine")
plt.show()
```

Python



Autres cas d'usage : analyser la répartition des évaluations clients d'un produit (5, 4, 3 étoiles...) ou celle des types de voitures (berline, SUV, camion) dans les données d'un concessionnaire.

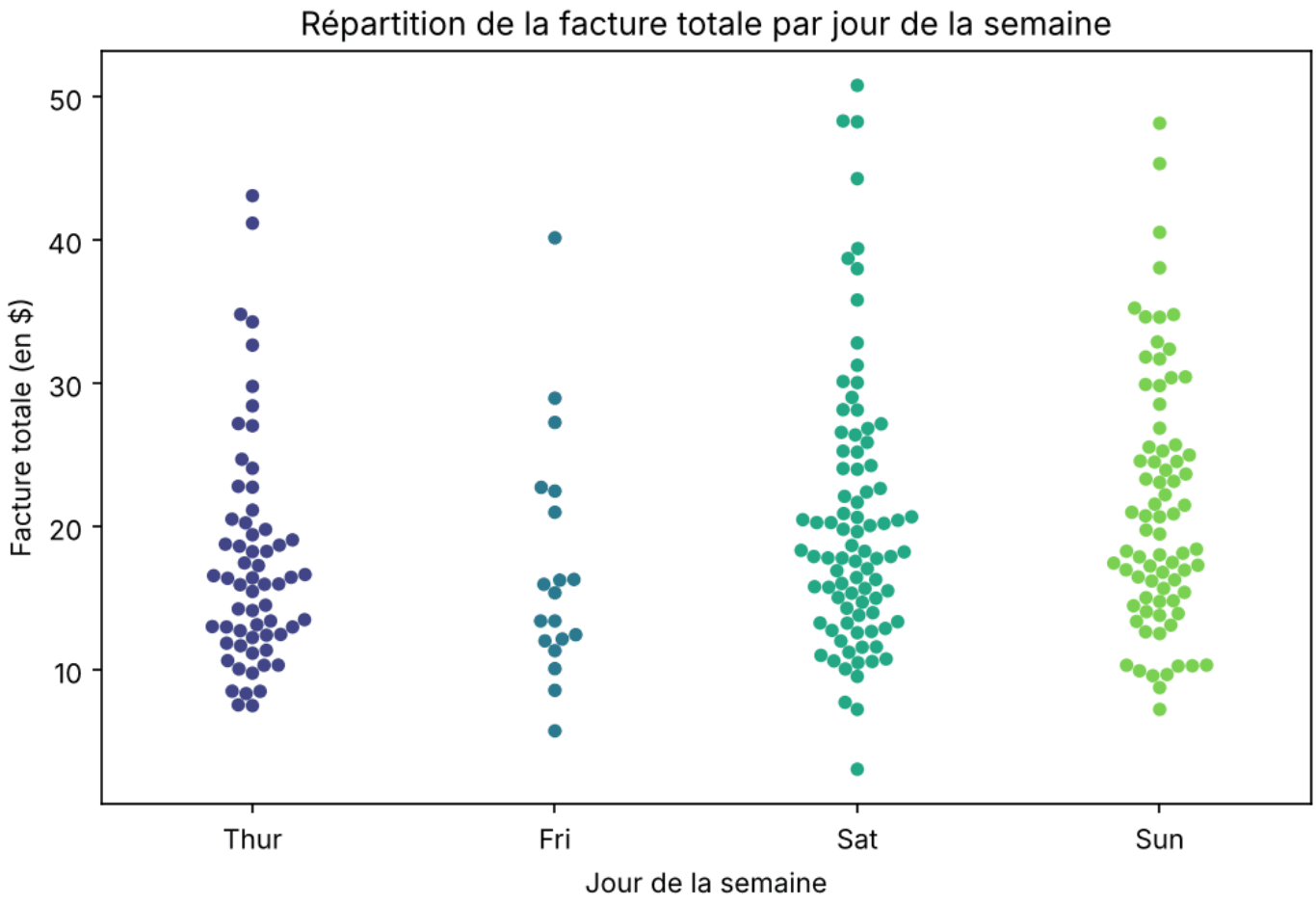
Swarm Plot

Un swarm plot (diagramme en essaim) a généralement une variable catégorielle sur l'axe x et une variable numérique sur l'axe y. Il affiche les points individuels de chaque catégorie, et permet donc de visualiser leur distribution et leur densité.

Sa principale caractéristique : il minimise le chevauchement des points. Chaque point est placé pour ne pas en recouvrir un autre de la même catégorie, ce qui rend la densité facile à lire. En revanche, il devient peu lisible sur de grands jeux de données : il est efficace sur des jeux relativement petits. Utilise `sns.swarmplot(x=..., y=..., data=...)` et `palette` pour changer les couleurs.

```python
plt.figure(figsize=(8, 5))
sns.swarmplot(x="day", y="total_bill", data=tips, palette="viridis", hue="day", legend=False)
plt.title("Répartition de la facture totale par jour de la semaine")
plt.xlabel("Jour de la semaine")
plt.ylabel("Facture totale (en $)")
plt.show()
```

Python



Un swarm plot aide à repérer les valeurs aberrantes ou les points inhabituels d'une catégorie, et à comprendre où les données sont concentrées ou clairsemées.

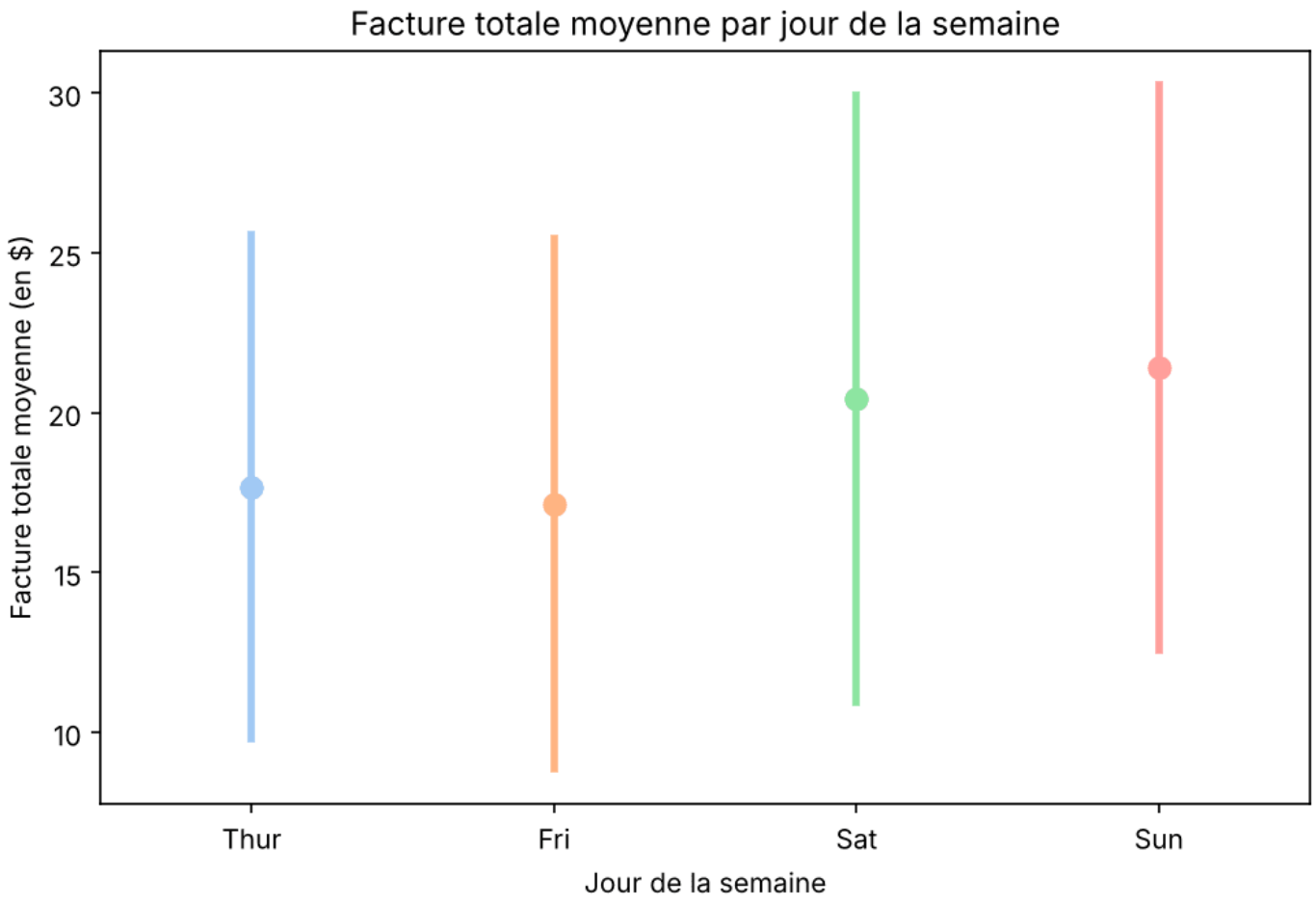
Autres cas d'usage : visualiser la répartition des notes d'examen entre plusieurs classes, ou celle des heures d'arrivée des clients selon le jour.

Point Plot

Un point plot a lui aussi une variable catégorielle sur l'axe x et une variable numérique sur l'axe y. Il représente, pour chaque catégorie, une tendance centrale (la moyenne par défaut) sous forme d'un point, avec une barre d'incertitude, et relie les points par une ligne. Tu vois rapidement s'il existe des différences entre les catégories.

```python
plt.figure(figsize=(8, 5))
sns.pointplot(x="day", y="total_bill", data=tips, errorbar="sd", palette="pastel", hue="day", legend=False)
plt.title("Facture totale moyenne par jour de la semaine")
plt.xlabel("Jour de la semaine")
plt.ylabel("Facture totale moyenne (en $)")
plt.show()
```

Python



Les cas d'usage sont les mêmes que pour le swarm plot, mais ici on observe la tendance centrale : on se fait une idée des scores moyens, faibles et élevés de chaque classe. Le paramètre `errorbar='sd'` affiche l'écart type ; par défaut, Seaborn affiche un intervalle de confiance à 95 %.

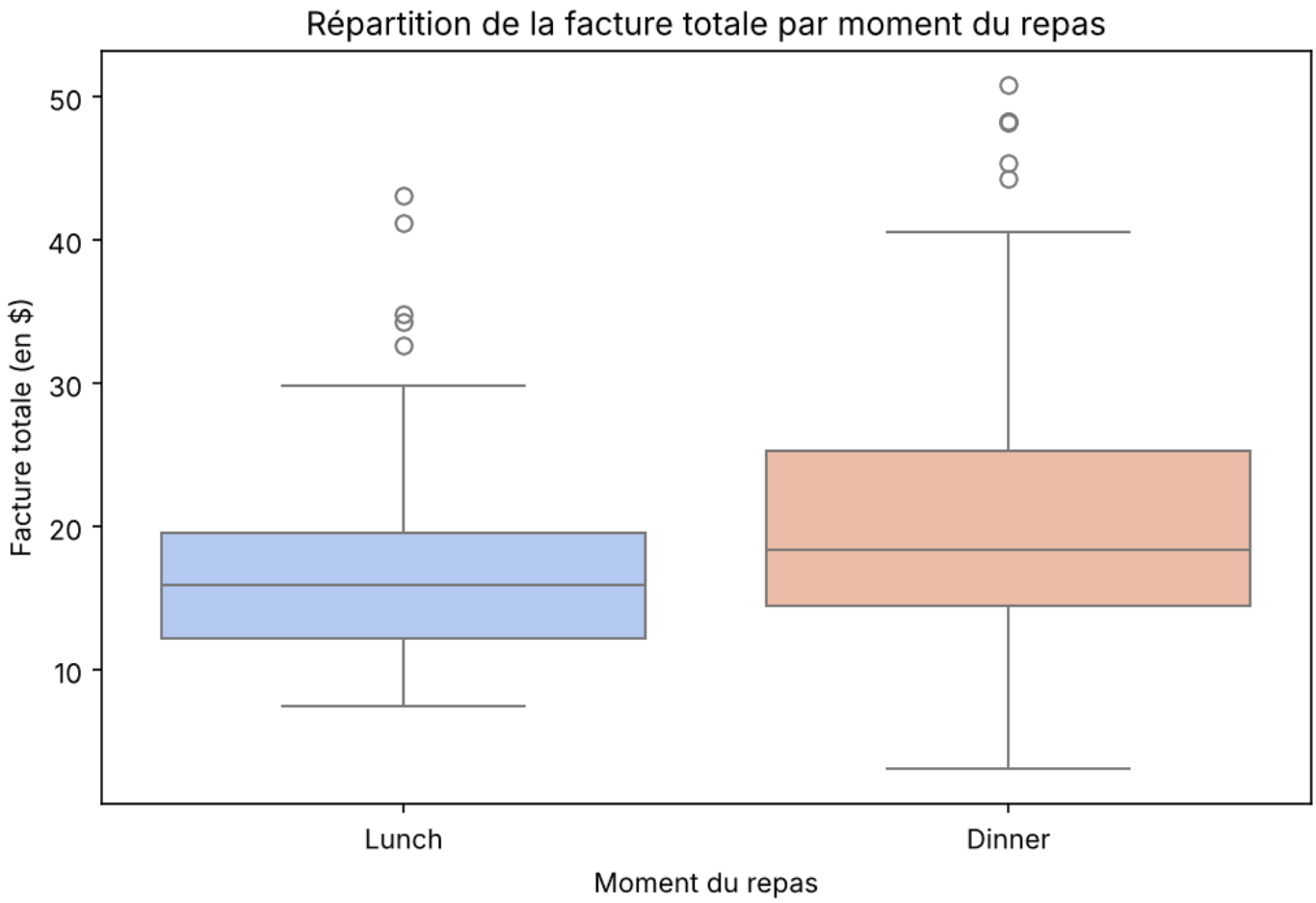
Box Plot catégoriel

Un box plot (diagramme en boîte, ou boîte à moustaches) visualise la distribution d'une variable numérique au sein de plusieurs catégories. Il affiche les quartiles (médiane, quartiles inférieur et supérieur), les valeurs aberrantes potentielles et la dispersion générale des données.

Utilise `sns.boxplot(x=..., y=..., data=...)`. Il n'est pas réservé aux catégories : tu peux aussi passer une colonne numérique à l'axe x.

```python
plt.figure(figsize=(8, 5))
sns.boxplot(x="time", y="total_bill", data=tips, palette="coolwarm", hue="time", legend=False)
plt.title("Répartition de la facture totale par moment du repas")
plt.xlabel("Moment du repas")
plt.ylabel("Facture totale (en $)")
plt.show()
```

Python



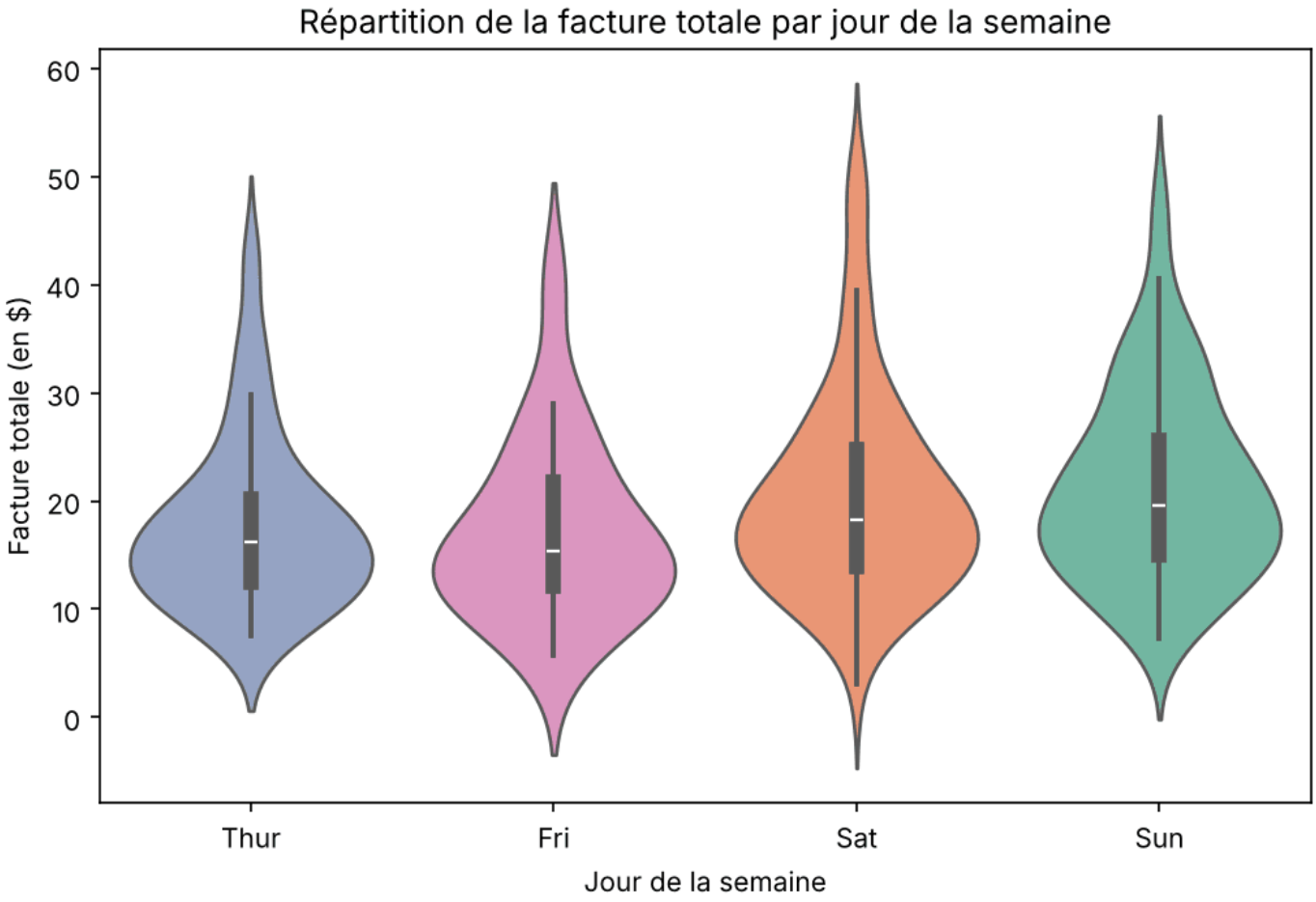
- La **boîte** couvre l'intervalle interquartile (IIQ), avec une ligne pour la médiane (50e percentile) ;
- Les **moustaches** s'étendent (souvent jusqu'à 1,5 fois l'IIQ) depuis la boîte. Les points situés au-delà sont des valeurs aberrantes potentielles ;
- Utilise-le pour obtenir des statistiques de base sur une variable catégorielle par rapport à une variable numérique. Essaie aussi avec deux variables numériques, comme `size` et `total_bill`.

Violin Plot catégoriel

La variable catégorielle détermine les groupes, et une variable numérique sert à dessiner un violon pour chaque groupe. La largeur du violon à une valeur donnée indique la densité des points à cette valeur. À l'intérieur, on trouve souvent un petit diagramme en boîte avec la médiane, les quartiles et les valeurs aberrantes potentielles. Utilise `sns.violinplot(x=..., y=..., data=...)`.

```
plt.figure(figsize=(8, 5))
sns.violinplot(x="day", y="total_bill", data=tips, palette="Set2", hue="day", legend=False)
plt.title("Répartition de la facture totale par jour de la semaine")
plt.xlabel("Jour de la semaine")
plt.ylabel("Facture totale (en $)")
plt.show()
```

Python



Le violin plot combine les caractéristiques d'un box plot et d'un diagramme de densité. Il est efficace pour comparer la distribution d'une variable numérique entre plusieurs catégories : tu vois non seulement la tendance centrale et la dispersion, mais aussi la forme et l'asymétrie de la distribution.

Cat Plot

Un cat plot (*categorical plot*) est une fonction très puissante : avec un seul paramètre, `kind`, dans la commande `sns.catplot(x=..., y=...)`, elle réalise tous les tracés catégoriels vus plus haut. Tu peux utiliser `kind='bar'`, `'swarm'`, `'box'`, `'violin'`, `'count'`, `'point'`, etc. C'est un véritable couteau suisse pour visualiser les relations entre variables catégorielles et numériques.

```python
sns.catplot(x="day", y="total_bill", data=tips, kind="box", height=5, aspect=1.5)
plt.show()

# Même appel, autre tracé : il suffit de changer kind
sns.catplot(x="day", y="total_bill", data=tips, kind="violin", height=5, aspect=1.5)
plt.show()
```

Python

La suite de la série porte sur les diagrammes univariés avec Seaborn. Documentation : visualiser des données catégorielles avec Seaborn.

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

