1. Analyse de la target

- Utiliser la bibliothèque pandas pour récupérer les données, vous supprimerez ensuite les 15 features suivantes :
  'id','description', 'first_review', 'host_has_profile_pic', 'host_identity_verified', 'host_response_rate', 'host_since', 'last_review', 'latitude', 'longitude', 'name','number_of_reviews', 'review_scores_rating', 'thumbnail_url', 'zipcode'
- Transformer la colonne *log_price* en une colonne *price*.
- Identifiez les locations pour lesquelles il n'y a pas d'information sur les commodités ('{}'). Supprimez les *features* correspondantes.
- Combien y a-t-il de doublons, supprimez ces doublons.
- Conserver maintenant dans la base de données seulement les locations comprises entre 20 et 1000 dollars, combien reste-t-il de *samples*.
- Afficher la répartition des prix avec un découpage en 100 bacs.
- Identifier les trois types de propriétés les plus importantes en nombre, afficher via *histplot* ces trois types de propriétés. Dans les quatre prochaines questions vous vous limiterez aux *samples* ayant un prix inférieur à 500 dollars. Que peut-on voir.
- Calculer maintenant la valeur moyenne de ces propriétés, que constate-t-on. Comment peut-on expliquer ces différences apparentes.
- Effectué maintenant un affichage avec trois *hisplot* sur des *dataframes* ne contenant qu'un seul type de propriété. Vous y positionnerez l'option *stat='percent'*. Que peut-on constater, est-ce que ce type d'affichage permet de mieux expliquer les résultats précédents.
- Que peut-on conclure si l'on donne trop d'importance sur la seule valeur des moyennes.

2. Traitement de *instant_bookable*

- Créer une liste contenant le nom de toutes les *features* objets. Pour chacune de ces *features* affichez le nombre de valeurs différentes.
- Combien y a-t-il de *samples* avec la valeur '*t*' ou '*f*' dans *instant_bookable,* quel est le prix moyen des locations de chacune de ces catégories.
- Y-a-t-il des données manquantes.

- Créer une *dataFrame* pour chacune des deux catégories grâce à la fonction *groupby*, vous ne traiterez que les locations de moins de 500 dollars.

  Info : Cette fonction retourne un objet que vous pouvez parcourir comme une liste composée de *tuple* (*valeur du groupe*, *dataframe*). Vous pouvez également récupérer chacune des *dataframes* via la méthode *get_group(valeur du groupe).*

- Afficher la via un histogramme la répartition des prix de chacune de ces catégories. Que constatez-vous, est-il judicieux de conserver cette *feature* ?

  Info : Vous utiliserez un affichage normalisé grâce à l'option *stat* de *histplot*.

3. Traitement de *room_type*

- Afficher le nombre et les différentes moyennes des différents types de chambre, vous n'afficherez que les locations à moins de 500 dollars. Que constatez-vous ?
- Au regard de ces premières informations est-il nécessaire de créer une *feature* pour chaque type de chambre, ou une seule colonne suffit.
- Affichez la variation des prix des locations avec la fonction *histplot*.
- Si l'on souhaite conserver une seule colonne, peut-on les numéroter dans un ordre aléatoire.
- Sachant que la plupart des algorithmes d'apprentissage utilisent la norme euclidienne pour comparer les samples entre eux, quel serait la meilleure façon de représenter ces différents types.

4. Analyse des *property_type*

- Combien y a-t-il de type de propriétés différentes et quel est le prix moyen de ces locations. Vous afficherez le résultat par ordre croissant du nombre du nombre de chaque propriété.

  Info : La fonction *aggregate()* combinée aux agrégations *groupby* permet d'appliquer plusieurs opérations classiques aux *dataframes* ou à une partie des *features*.

- Pensez-vous qu'il soit nécessaire de conserver toutes ces catégories.
- On souhaite regrouper tous les types de propriétés qui ont un prix moyen élevé > 200 dollars. Créer une liste correspondant à ces types de propriétés et transformez leur valeur en *Timeshare*.
- Créer une liste contenant les types de propriétés avec moins de 20 valeurs, et supprimez-les.

Info : Vous pouvez utiliser la fonction *isin()*.

5. Analyse des *bed_type*

- Afficher le nombre de locations par types de lits ainsi que le prix moyen. Quelle analyse peut-on faire de ce premier résultat.
- Afin d'avoir des informations plus détaillées, on cherche les prix moyens des locations en fonction des types de propriétés et des types de lits. Calculer le prix moyen en fonction de ces deux *features*.

  Info : Vous utiliserez la fonction *groupby* sur les deux *features property_type* et *bed_type*.
- Modifier le *dataframe* obtenue pour en faire une lecture plus simple.

  Info : Sur le résultat précédent la fonction pivot va permettre de réorganiser le dataframe. Pour cela vous choisirez comme *index* la colonne *property_type*, comme colonnes la colonne *bed_type* et comme valeurs les prix moyens obtenus *price*.
- Effectuer le même travail sur le nombre de locations sur la combinaison des deux *features*.
- Que constatez-vous concernant en particulier les types *Other* et *loft*, que peut-on en conclure. Voyez-vous une *feature* que semble avoir un impact fort sur les prix des locations.
- Peut-on numériser ces données dans n'importe quel ordre ou faut-il conserver une certaine hiérarchie entre les différents types de lits.

6. Traitement de *accommodates*

- Afficher le nombre de locations et le prix moyen des locations en fonction de la *feacture accommodates*. Que constatez-vous ?
- Analyser plus finement le prix moyen des locations lorsque le nombre de salle de bains est égal à 8. Comment peut-on expliquer cela.
- Pourquoi est-il plus judicieux d'encoder les valeurs en fonction des prix moyens des locations plutôt que sur le nombre.

7. Numérisation des *feature*

- Combien y a-t-il de valeur *Nan* dans les différentes *features*. Supprimer toutes ces valeurs de la base.
- Avant toutes transformations effectuer une copie de la *dataframe*.

- Transformez la *feacture room_type* en données numériques basées sur leur prix moyen. Vous utiliserez la fonction *replace*.
- Transformez également la *feacture bed_type* de la même manière.
- Afficher le nombre de locations et le prix moyen des locations en fonction de la *feacture cancelation_policy*. Numériser cette *feature* sur la base des prix moyens.
- Afficher le nombre de locations et le prix moyen des locations en fonction de la *feacture cleaning_fee*. Numériser cette *feature* sur la base des prix moyens.
- Numériser les *feature bedrooms, beds, bathrooms, property_type, accommodates* en fonction de leurs prix moyens.
- Supprimer de la base les données les *features* de type *object* restantes.

8. Identification d'outliers

- Utiliser un modèle *IsolationForest* pour identifier les *outliers* (avec une contamination de 1/1000).
- A partir des valeurs d'origine sauvegardées dans la copie du *dataframe* avant la numérisation rechercher comment se répartissent les *outliers*.
- Quels sont les principaux critères qui permettent de les identifier.