TP Pandas -

Data

Cet ensemble de données contient des informations de réservation pour un hôtel urbain et un hôtel de villégiature, et comprend des informations telles que la date à laquelle la réservation a été effectuée, la durée du séjour, le nombre d'adultes, d'enfants et/ou de bébés et le nombre de places de stationnement disponibles, entre autres.

Toutes les informations d'identification personnelle ont été supprimées des données.

NOTE: Les noms, e-mails, numéros de téléphone et numéros de carte de crédit contenus dans les données sont des informations synthétiques et non réelles provenant de personnes. Les données de l'hôtel sont réelles et proviennent de la publication répertoriée ci-dessous.

Cours : DATA MANAGEMENT, DATAVIZ, TEXT MINING

Élève : Kenny NUNGU (Formation initiale)

Variable	Type	Description
ADR	Numérique	Tarif journalier moyen tel que défini par [5]
AdultsChildBaby	Entier	Nombre d'adultes
Agent	Catégorique	ID de l'agence de voyage ayant effectué la réservation[1]
ArrivalDateDayOfMonth(JourDuMoisArrivee	Entier	Jour du mois de la date d'arrivée
ArrivalDateMonth(MoisArrivee	Catégorique	Mois de la date d'arrivée avec 12 catégories : "Janvier" à "Décembre"
ArrivalDateWeekNumber(NumeroSemaineArrivee	Entier	Numéro de semaine de la date d'arrivée
ArrivalDateYear(AnneeArrivee	Entier	Année de la date d'arrivée
AssignedRoomType(TypeChambreAttribuee	Catégorique	Code du type de chambre attribuée à la réservation. Parfois, le type de chambre attribué diffère du type de chambre réservé pour des raisons opérationnelles du (hôtel (par exemple, surréservation) ou à la demande du client. Le code est présenté au lieu du (hôtel pour des raisons d'anonymat
Babies(Bébés	Entier	Nombre de bébés
BookingChanges(ModificationsReservation	Entier	Nombre de modifications/amendements apportés à la réservation depuis le moment où la réservation a été saisie dans le PMS jusqu'au moment de l'enregistrement ou de l'annulation
Children(Enfants	Entier	Nombre d'enfants
Company(SocieteE	Catégorique	ID de la société/entité ayant effectué la réservation ou responsable du paiement de la réservation. L'ID est présenté au lieu du (hôtel pour des raisons d'anonymat
Country(Pays	Catégorique	Pays d'origine. Les catégories sont représentées dans le format ISO 3155-3:2013 [6]
CustomerType(TypeClient	Catégorique	Type de réservation, en supposant l'une des quatre catégories : Contrat - lorsque la réservation est associée à un autre type de contrat ; Groupe - lorsque la réservation est associée à un groupe ; Transitoire - lorsque la réservation ne fait partie d'un groupe ou d'un contrat, et n'est pas associée à d'autres réservations transitoires ; Groupe Transitoire - lorsque la réservation est transitoire, mais est associée à au moins une autre réservation transitoire
DaysInWaitingList(JoursEnListeAttente	Entier	Nombre de jours pendant lesquels la réservation était en liste d'attente avant d'être confirmée au client
DepositType(TypeDepot	Catégorique	Indication de savoir si le client a versé un dépôt pour garantir la réservation. Cette variable peut prendre trois catégories : Pas de Dépôt - aucun dépôt n'a été effectué ; / Si aucun paiement n'a été trouvé, la valeur est "Pas de Dépôt". Si le paiement était égal ou supérieur au coût total du séjour, il est défini comme "Non Remboursable" ; Non Remboursable - un dépôt a été effectué pour couvrir la valeur intégrale du coût total du séjour. / Sinon, la valeur est définie comme "Remboursable"
DistributionChannel(CanalDistribution	Catégorique	Canal de distribution de la réservation. Le terme "TA" signifie "Agences de Voyage" et "TO" signifie "Tour-Opérateurs"
IsCanceled(EstAnnule	Catégorique	Valeur indiquant si la réservation a été annulée (1) ou non (0)
IsRepeatedGuest(EstClientHabitue	Catégorique	Valeur indiquant si le nom de la réservation est celui d'un client régulier (1) ou non (0)
LeadTime(DelaiAvantArrivee	Entier	Nombre de jours écoulés entre la date d'entrée de la réservation dans le PMS et la date d'arrivée
MarketSegment(SegmentMarché	Catégorique	Désignation du segment de marché. Dans les catégories, le terme "TF" signifie "Agences de Voyage" et "TO" signifie "Tour-Opérateurs"
Meal(Repas	Catégorique	Type de repas réservé. Les catégories sont présentées dans les forfaits repas standard de l'hôtelerie : Non DefiniDA - aucun forfait repas ; FD - Petit-Déjeuner ; Demi-Pension (petit-déjeuner et un autre repas - généralement le dîner) ; Pension Complète (petit-déjeuner, déjeuner et dîner)
PreviousBookingsNotCanceled(ReservationsPrecedentesNonAnnulees	Entier	Nombre de réservations précédentes non annulées par le client avant la réservation actuelle
PreviousCancellations(ReservationsPrecedentesAnnulees	Entier	Nombre de réservations précédentes annulées par le client avant la réservation actuelle
RequiredCarParkingSpaces(PlacesParkingRequises	Entier	Nombre de places de parking requises par le client
ReservationStatus(StatutReservation	Catégorique	Dernier statut de la réservation, en supposant l'une des trois catégories : Annulée - la réservation a été annulée par le client ; Check-Out - le client s'est enregistré mais est déjà parti ; No-Show - le client n'est pas venu et n'a pas informé l'hôtel de la raison
ReservationStatusDate(DateStatutReservation	Date	Date à laquelle le dernier statut a été défini. Cette variable peut être utilisée conjointement avec le (StatutReservation pour comprendre quand la réservation a été annulée ou quand le client a effectué le check-out de l'hôtel
ReservedRoomType(TypeChambreReservé	Catégorique	Code du type de chambre réservée. Le code est présenté au lieu du (hôtel pour des raisons d'anonymat
StaysInWeekendNights(NuitsSemaineSejournees	Entier	Nombre de nuits du week-end (samedi ou dimanche) pendant lesquelles le client a séjourné ou a réservé pour séjourner à l'hôtel
StaysInWeekNights(NuitsSemaineSejournees	Entier	Nombre de nuits en semaine (du lundi au vendredi) pendant lesquelles le client a séjourné ou a réservé pour séjourner à l'hôtel
TotalOfSpecialRequests(TotalDemandesSpéciales	Entier	Nombre de demandes spéciales faites par le client (par exemple, lit jumeau ou étage élevé)

In [1771]:
```
import pandas as pd
```

In [1772]:
```
hotels = pd.read_csv('Users/kennynungu/Downloads/hotel_booking_data.csv')
```

In [1773]:
```
hotels['country'].value_counts()
```

Out[1773]:
```
country
PRT    48590
GBR    12129
FRA    10415
ESP     8568
DEU     7287
       ...  
BJI        1
DMA        1
PLW        1
VGB        1
NAM        1
Name: count, Length: 177, dtype: int64
```

In [1883]:
```
hotels.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 36 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   hotel                           119390 non-null  object 
 1   is_canceled                     119390 non-null  int64  
 2   lead_time                       119390 non-null  int64  
 3   arrival_date_year               119390 non-null  int64  
 4   arrival_date_month              119390 non-null  object 
 5   arrival_date_week_number        119390 non-null  int64  
 6   arrival_date_day_of_month       119390 non-null  int64  
 7   stays_in_weekend_nights         119390 non-null  int64  
 8   stays_in_week_nights            119390 non-null  int64  
 9   adults                          119390 non-null  int64  
 10  children                        119386 non-null  float64
 11  babies                          119390 non-null  int64  
 12  meal                            119390 non-null  object 
 13  country                         118902 non-null  object 
 14  market_segment                  119390 non-null  object 
 15  distribution_channel            119390 non-null  object 
 16  is_repeated_guest               119390 non-null  int64  
 17  previous_cancellations          119390 non-null  int64  
 18  previous_bookings_not_canceled  119390 non-null  int64  
 19  reserved_room_type              119390 non-null  object 
 20  assigned_room_type              119390 non-null  object 
 21  booking_changes                 119390 non-null  int64  
 22  deposit_type                    119390 non-null  object 
 23  agent                           103050 non-null  float64
 24  company                         6797 non-null    float64
 25  days_in_waiting_list            119390 non-null  int64  
 26  customer_type                   119390 non-null  object 
 27  adr                             119390 non-null  float64
 28  required_car_parking_spaces     119390 non-null  int64  
 29  total_of_special_requests       119390 non-null  int64  
 30  reservation_status              119390 non-null  object 
 31  reservation_status_date         119390 non-null  object 
 32  name                            119390 non-null  object 
 33  email                           119390 non-null  object 
 34  phone-number                    119390 non-null  object 
 35  credit_card                     119390 non-null  object 
dtypes: float64(4), int64(16), object(16)
memory usage: 32.8+ MB
```

Combien y a-t-il de lignes ?

In [1884]:
```
hotels.shape[0]
```

Out[1884]:
```
119390
```

Y a-t-il des données manquantes ? Si oui, quelle colonne contient le plus de données manquantes ?

In [1885]:
```
if hotels.isnull().values.any():
    print("Il y a des données manquantes.")
else:
    print("Il n'y a pas de données manquantes.")

# Nombre de données manquantes par colonne
données_manquantes = hotels.isnull().sum()

# Colonne qui contient le plus de données manquantes
colonne_données_manquantes = données_manquantes.idxmax()
nombre_données_manquantes = données_manquantes.max()

print("Il y a des données manquantes.")
```

```
Il y a des données manquantes.
```

In [1886]:
```
print("Plus", la colonne qui contient le plus de données est : '{colonne_données_manquantes}' avec {nombre_données_manquantes} données manquantes.")
```

```
Plus, la colonne qui contient le plus de données est : 'company' avec 112593 données manquantes.
```

Supprimez la colonne « company » de l'ensemble de données.

In [1255]:
```
hotels = hotels.drop('company',axis=1)
# on pourrait aussi utiliser inplace=True donc hotels.drop('company', axis1,inplace=True) au encore hotels.dropicolumns='company', inplace=True)
```

Supprimez les lignes qui n'ont pas de valeurs pour la colonne 'children'

In [1885]:
```
hotels.dropna(subset=['children'])
```

Out[1885]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults
0	Resort Hotel	0	342	2015	July	27	1	0	0	2
1	Resort Hotel	0	737	2015	July	27	1	0	0	2
2	Resort Hotel	0	7	2015	July	27	1	0	1	1
3	Resort Hotel	0	13	2015	July	27	1	0	1	1
4	Resort Hotel	0	14	2015	July	27	1	0	2	2
...	...	...	...	...	...	...	...	...	...	...
119386	City Hotel	0	23	2017	August	35	30	2	5	2
119388	City Hotel	0	102	2017	August	35	31	2	5	3
119387	City Hotel	0	34	2017	August	35	31	2	5	2
119389	City Hotel	0	109	2017	August	35	31	2	5	2
119390	City Hotel	0	205	2017	August	35	29	2	7	2

119386 rows × 35 columns

Remplir les valeurs manquantes de la colonne 'country' par la valeur la plus fréquente (la valeur mode de la colonne)

In [1886]:
```
# Calcul de la valeur modale plus fréquente de la colonne 'country'
valeur_plus_fréquente_country = hotels['country'].mode()[0]

# Remplissage des valeurs manquantes par la valeur modale plus fréquente
hotels['country'].fillna(valeur_plus_fréquente_country)
```

Out[1886]:
```
0        PRT
1        PRT
2        GBR
3        GBR
4        GBR
        ... 
119385   FRA
119386   FRA
119388   GBR
119389   DEU
Name: country, Length: 119390, dtype: object
```

Affichez les 5 pays les plus fréquents de la colonnes 'country'

In [1888]:
```
pays_les_plus_fréquents = hotels['country'].value_counts().head(5)
pays_les_plus_fréquents
```

Out[1888]:
```
country
PRT    48590
GBR    12129
FRA    10415
ESP     8568
DEU     7287
Name: count, dtype: int64
```

Quel est le nom de la personne qui a payé l'ADR (taux journalier moyen) le plus élevé ? Quel était son ADR ?

In [1237]:
```
# On cherche d'abord l'index de la ligne avec l'ADR le plus élevé
adr_le_plus_élevé = hotels['adr'].idxmax()

# On cherche le nom de la personne et son ADR correspondant
nom_de_la_personne = hotels.loc[adr_le_plus_élevé, 'name']
valeur_adr_max = hotels.loc[adr_le_plus_élevé, 'adr']

print(f"Le nom de la personne ayant payé l'ADR le plus élevé est : {nom_de_la_personne} avec un ADR de {valeur_adr_max}.")
```

```
Le nom de la personne ayant payé l'ADR le plus élevé est : Daniel Walter avec un ADR de 5400.0.
```

L'ADR est le tarif journalier moyen pour le séjour d'une personne à l'hôtel. Quelle est l'ADR moyen pour tous les séjours hôtels de l'ensemble de données ?

In [1241]:
```
hotels['adr'].mean()
```

Out[1241]:
```
101.83122534466696
```

Quel est le nombre moyen de nuits pour un séjour sur l'ensemble des données ?

A noter qu'il faut prendre en considération les deux colonnes stays_in_week_nights & stays_in_weekend_nights

In [1242]:
```
# On calcule le nombre total de nuits pour chaque séjour
hotels['nombre_total_nuits'] = hotels['stays_in_week_nights'] + hotels['stays_in_weekend_nights']

# On calcule le nombre moyen de nuits
nombre_moyen_nuits = hotels['nombre_total_nuits'].mean()

print(f"Le nombre moyen de nuits pour un séjour sur l'ensemble des données est : {nombre_moyen_nuits:.2f}")
```

```
Le nombre moyen de nuits pour un séjour sur l'ensemble des données est : 3.43
```

Quel est le coût total moyen d'un séjour dans l'ensemble de données ? Pas le coût journalier moyen, mais le coût total du séjour.

In [1243]:
```
# On calcule le coût total de chaque séjour
hotels['cout_total'] = hotels['adr'] * hotels['nombre_total_nuits']

# Puis le coût total moyen d'un séjour
cout_total_moyen = hotels['cout_total'].mean()

print(f"Le coût total moyen d'un séjour est : {cout_total_moyen:.2f}")
```

```
Le coût total moyen d'un séjour est : 351.85
```

Quels sont les noms et adresses e-mail des personnes qui ont fait 5 « special_requests » ?

In [1244]:
```
cinq_special_requests = hotels[hotels['total_of_special_requests'] == 5][['name', 'email']]

print("Les noms et adresses e-mail des personnes ayant fait 5 special_requests :")
cinq_special_requests
```

```
Les noms et adresses e-mail des personnes ayant fait 5 special_requests :
```

Out[1244]:

	name	email
7880	Amanda Harper	Amanda.H@outlook.com
11725	Laura Sanders	Sanders_Laura@hotmail.com
14596	Tommy Ortiz	Tommy_O@hotmail.com
14621	Gilbert Miller	Miller.Gilbert@aol.com
14922	Timothy Torres	TTorres@protonmail.com
24830	Jennifer Weaver	Jennifer_W@aol.com
37368	Crystal Horton	Crystal.H@mail.com
37477	Brittney Burke	Burke_Brittney1@aol.com
39906	Cynthia Cabrera	Cabrera.Cynthia@outlook.com
39949	Sarah Floyd	Sarah_F@gmail.com
53367	Michelle Villa	Michelle.V@aol.com
39027	Nicholas Hebert	Hebert.Nicholas@gmail.com
39129	Lindsey Mckenzie	Lindsey.Mckenzie@att.com
39525	Ashley Edwards	Edwards.Ashley@yahoo.com
70114	Christopher Torres	Torres.Christopher@gmail.com
78898	Mrs. Tara Sullivan DVM	Mrs._DVM@infinity.com
79020	Michaela Brown	MichaelaB@outlook.com
78822	Kurt Maldonado MD	KMD1@infinity.com
91972	Jason Richardson	Jason.R@otlook.com
97099	Terri Hurley	ThurleyTer@infinity.com
97261	Mrs. Caitlin Webb	Mrs._W@comcast.net
98410	Holly Arroyo	Arroyo_Holly@icloud.com
98674	Denise Campbell	Denise_C@gmail.com
99887	Michael Smith	Michael.S423@aol.com
99888	Dr. Trevor Sellers	Dr._S@aol.com
101069	Kayla Murphy	Kayla.Murphy@yahoo.com
102581	Taylor Martinez	Taylor.Martinez@hotmail.com
109511	Charles Wilson	Charles_Wilson@yahoo.com
109590	Tyler Allison	Tyler.A@protonmail.com
110082	Matthew Bailey	Matthew_Bailey@aol.com
110083	Charlotte Acevedo	Charlotte_A@verizon.com
111909	Darrell Brennan	Brennan_Darrell1@hotmail.com
111911	Melinda Jensen	MelindaJensen@cnbc.com
112915	Terry Arnold	Arnold.Terry@outlook.com
114770	Mary Nguyen	Nguyen.Mary@protonmail.com
114909	Lindsay Cuevas	Lindsay.Cuevas93@mail.com
116456	Cynthia Hernandez	Cynthia.Hernandez@yahoo.com
116457	Angela Hawkins	Angela_H@gmail.com
116817	Sue Lawson	Sue_L52@comcast.net
119561	Alyssa Richards	Alyssa_Richards@aol.com

Quel est le pourcentage de séjours à l'hôtel qui ont été classés comme « clients réguliers » ? (Ne basez pas cela sur le nom de la personne, mais plutôt sur la colonne is_repeated_guest)

In [1245]:
```
clients_reguliers_pourcentage = hotels['is_repeated_guest'].mean() * 100

print(f"Le pourcentage de séjours à l'hôtel ayant été classés comme « clients réguliers » est de : {clients_reguliers_pourcentage:.2f}%")
```

```
Le pourcentage de séjours à l'hôtel ayant été classés comme « clients réguliers » est de : 3.19%
```

Quels sont les noms des 3 personnes qui ont réservé le plus grand nombre d'enfants et de bébés pour leur séjour ? (Ne vous inquiétez pas s'ils annulent, considérez uniquement le nombre de personnes signalé au moment de leur réservation)

In [1246]:
```
# Calcul du nombre total d'enfants et de bébés par réservation
hotels['enfants_et_bebes'] = hotels['children'] + hotels['babies']

# On trie ensuite les réservations en fonction du total d'enfants et de bébés
trié_réservations = hotels.sort_values(by='enfants_et_bebes', ascending=False)

# On obtient les noms des 3 personnes
trois_personnes = trié_réservations.head(3)[['name', 'enfants_et_bebes']]

print("Le nom des 3 personnes ayant réservé pour le plus grand nombre d'enfants et de bébés pour leur séjour :")
trois_personnes
```

```
Le nom des 3 personnes ayant réservé pour le plus grand nombre d'enfants et de bébés pour leur séjour :
```

Out[1246]:

	name	enfants_et_bebes
328	Jamie Ramirez	10.0
40618	Nicholas Parker	10.0
70056	Marc Robinson	9.0

Quels sont les trois indicatifs régionaux les plus courants dans les numéros de téléphone ? (L'indicatif régional correspond aux 3 premiers chiffres). Bonus : pouvez-vous faire cela en une seule ligne de code pandas en utilisant apply & lambda function

In [1247]:
```
top_3_indicatifs_regionaux = hotels['phone-number'].apply(lambda phone: phone[:3]).value_counts().head(3).reset_axis('index')
top_3_indicatifs_regionaux
# on a utilisé reset_axis('index') à la fin de code pour plus de clarté
```

Out[1247]:
```
index
796    168
185    167
541    166
Name: count, dtype: int64
```

Combien d'arrivées ont eu lieu entre le 1er et le 15 du mois (1 et 15 inclus) ? Bonus : pouvez-vous faire cela en une seule ligne de code pandas en utilisant apply & lambda function

In [1248]:
```
arrivees_premier_quinze_mois = hotels['arrival_date_day_of_month'].apply(lambda arrivees1erMois: 1 <= arrivees1erMois <= 15).sum()
arrivees_premier_quinze_mois
```

Out[1248]:
```
56152
```