Pandas Project Exercise - Solutions

The Data

This data set contains booking information for a city hotel and a resort hotel, and includes information such as when the booking was made, length of stay, the number of adults, children, and/or babies, and the number of available parking spaces, among other things.

All personally identifying information has been removed from the data.

Acknowledgements The data is originally from the article Hotel Booking Demand Datasets, written by Nuno Antonio, Ana Almeida, and Luis Nunes for Data in Brief, Volume 22, February 2019.

NOTE: Names, Emails, Phone Numbers, and Credit Card numbers in the data are synthetic and not real information from people. The hotel data is real from the publication listed above.

Data Column Reference

TASKS

** Complete the tasks shown in bold below. The expected output is shown in a cell below. Be careful not to run the cell above the expected output, as it will clear the expected output. Try your best to solve these in one line of pandas code (not every single question can be solved in one line, but many can be!) Refer to solutions notebook and video to view possible solutions. NOTE: Many tasks have multiple correct solution methods!**

```
import pandas as pd
```

```
hotels = pd.read_csv("hotel_booking_data.csv")
```

```
hotels.head()
```

```
          hotel  is_canceled  lead_time  arrival_date_year
arrival_date_month  \
0  Resort Hotel            0        342               2015
July
1  Resort Hotel            0        737               2015
July
2  Resort Hotel            0          7               2015
```

```
July
3  Resort Hotel            0         13               2015
July
4  Resort Hotel            0         14               2015
July

   arrival_date_week_number  arrival_date_day_of_month  \
0                        27                          1
1                        27                          1
2                        27                          1
3                        27                          1
4                        27                          1

   stays_in_weekend_nights  stays_in_week_nights  adults  ...
customer_type  \
0                        0                     0       2  ...
Transient
1                        0                     0       2  ...
Transient
2                        0                     1       1  ...
Transient
3                        0                     1       1  ...
Transient
4                        0                     2       2  ...
Transient

    adr required_car_parking_spaces total_of_special_requests  \
0   0.0                           0                         0
1   0.0                           0                         0
2  75.0                           0                         0
3  75.0                           0                         0
4  98.0                           0                         1

  reservation_status reservation_status_date            name  \
0          Check-Out              2015-07-01   Ernest Barnes
1          Check-Out              2015-07-01    Andrea Baker
2          Check-Out              2015-07-02  Rebecca Parker
3          Check-Out              2015-07-02    Laura Murray
4          Check-Out              2015-07-03     Linda Hines

                         email  phone-number       credit_card
0  Ernest.Barnes31@outlook.com  669-792-1661  ************4322
1       Andrea_Baker94@aol.com  858-637-6955  ************9157
2   Rebecca_Parker@comcast.net  652-885-2745  ************3734
3            Laura_M@gmail.com  364-656-8427  ************5677
4          LHines@verizon.com   713-226-5883  ************5498

[5 rows x 36 columns]

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
```

TASK: How many rows are there?

```
# CODE HERE

len(hotels) #hotels.info()
```

```
119390
```

TASK: Is there any missing data? If so, which column has the most missing data?

```
# CODE HERE

# hotels.isna().sum() #works as well
hotels.isnull().sum()
```

```
hotel                               0
is_canceled                         0
lead_time                           0
arrival_date_year                   0
arrival_date_month                  0
arrival_date_week_number            0
arrival_date_day_of_month           0
stays_in_weekend_nights             0
stays_in_week_nights                0
adults                              0
children                            4
babies                              0
meal                                0
country                           488
market_segment                      0
distribution_channel                0
is_repeated_guest                   0
previous_cancellations              0
previous_bookings_not_canceled      0
reserved_room_type                  0
assigned_room_type                  0
booking_changes                     0
deposit_type                        0
agent                           16340
company                        112593
days_in_waiting_list                0
customer_type                       0
adr                                 0
required_car_parking_spaces         0
total_of_special_requests           0
reservation_status                  0
reservation_status_date             0
name                                0
email                               0
phone-number                        0
credit_card                         0
dtype: int64
```

```
print(f"Yes, missing data, company column missing:
{hotels['company'].isna().sum()} rows.")
```

```
Yes, missing data, company column missing: 112593 rows.
```

TASK: Drop the "company" column from the dataset.

```
# CODE HERE
```

```
hotels = hotels.drop('company',axis=1)
```

TASK: What are the top 5 most common country codes in the dataset?

```
# CODE HERE
```

```
hotels['country'].value_counts()[:5]
```

```
PRT    48590
GBR    12129
FRA    10415
ESP     8568
DEU     7287
Name: country, dtype: int64
```

TASK: What is the name of the person who paid the highest ADR (average daily rate)? How much was their ADR?

```
# CODE HERE
```

```
hotels.sort_values('adr',ascending=False)[['adr','name']].iloc[0]
```

```
adr              5400
name    Daniel Walter
Name: 48515, dtype: object
```

TASK: The adr is the average daily rate for a person's stay at the hotel. What is the mean adr across all the hotel stays in the dataset?

```
# CODE HERE
```

```
round(hotels['adr'].mean(),2)
```

```
101.83
```

TASK: What is the average (mean) number of nights for a stay across the entire data set? Feel free to round this to 2 decimal points.

```
# CODE HERE
```

```
hotels['total_stay_days'] = hotels['stays_in_week_nights'] +
hotels['stays_in_weekend_nights']

round(hotels['total_stay_days'].mean(),2)
```

```
3.43
```

TASK: What is the average total cost for a stay in the dataset? Not *average daily cost*, but *total* stay cost. Feel free to round this to 2 decimal points.

```
# CODE HERE

hotels['total_paid'] = hotels['adr'] * hotels['total_stay_days']
round(hotels['total_paid'].mean(),2)
```

```
357.85
```

TASK: What are the names and emails of people who made 5 "Special Requests"?

```
# CODE HERE

hotels[hotels['total_of_special_requests'] == 5][['name','email']]
```

	name	email
7860	Amanda Harper	Amanda.H66@yahoo.com
11125	Laura Sanders	Sanders_Laura@hotmail.com
14596	Tommy Ortiz	Tommy_O@hotmail.com
14921	Gilbert Miller	Miller.Gilbert@aol.com
14922	Timothy Torres	TTorres@protonmail.com
24630	Jennifer Weaver	Jennifer_W@aol.com
27288	Crystal Horton	Crystal.H@mail.com
27477	Brittney Burke	Burke_Brittney16@att.com
29906	Cynthia Cabrera	Cabrera.Cynthia@xfinity.com
29949	Sarah Floyd	Sarah_F@gmail.com
32267	Michelle Villa	Michelle.Villa@aol.com
39027	Nichole Hebert	Hebert.Nichole@gmail.com
39129	Lindsey Mckenzie	Lindsey.Mckenzie@att.com
39525	Ashley Edwards	Edwards.Ashley@yahoo.com
70114	Christopher Torres	Torres.Christopher@gmail.com
78819	Mrs. Tara Sullivan DVM	Mrs..DVM@xfinity.com
78820	Michaela Brown	MichaelaBrown@att.com
78822	Kurt Maldonado MD	KMD15@xfinity.com
97072	Jason Richardson	Jason.R@zoho.com
97099	Terri Hurley	THurley@xfinity.com
97261	Mrs. Caitlin Webb	Mrs._W@comcast.net
98410	Holly Arroyo	Arroyo_Holly@mail.com
98674	Denise Campbell	Denise_C@gmail.com
99887	Michael Smith	Michael.S42@aol.com
99888	Dr. Trevor Sellers	Dr._S@aol.com
101569	Kayla Murphy	Kayla.Murphy@yahoo.com
102061	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_Darrell51@hotmail.com
111911	Melinda Jensen	MelindaJensen@zoho.com
113915	Terry Arnold	Arnold.Terry@zoho.com
114770	Mary Nguyen	Nguyen.Mary@protonmail.com
114909	Lindsay Cuevas	Lindsay.Cuevas40@mail.com
116455	Cynthia Hernandez	CynthiaHernandez@xfinity.com

```
116457          Angela Hawkins            Angela_H@gmail.com
118817              Sue Lawson           Sue.L52@comcast.net
119161        Alyssa Richards       Alyssa_Richards@aol.com
```

TASK: What percentage of hotel stays were classified as "repeat guests"? (Do not base this off the name of the person, but instead of the is_repeated_guest column)

```
#CODE HERE

# You can sum booleans, False gets treated as zero, True as one
round(100 * sum(hotels['is_repeated_guest'] == 1) / len(hotels),2)
```

```
3.19
```

TASK: What are the top 5 most common last name in the dataset? Bonus: Can you figure this out in one line of pandas code? (For simplicity treat the a title such as MD as a last name, for example Caroline Conley MD can be said to have the last name MD)

```
#CODE HERE

hotels['name'].apply(lambda name: name.split()[-1]).value_counts()[:5]
```

```
Smith       2503
Johnson     1990
Williams    1618
Jones       1434
Brown       1423
Name: name, dtype: int64
```

TASK: What are the names of the people who had booked the most number children and babies for their stay? (Don't worry if they canceled, only consider number of people reported at the time of their reservation)

```
#CODE HERE

hotels['total_kids'] = hotels['babies'] + hotels['children']

hotels.sort_values('total_kids',ascending=False)
[['name','adults','total_kids','babies','children']][:3]
```

```
                  name  adults  total_kids  babies  children
328      Jamie Ramirez       2        10.0       0      10.0
46619  Nicholas Parker       2        10.0      10       0.0
78656    Marc Robinson       1         9.0       9       0.0
```

TASK: What are the top 3 most common area code in the phone numbers? (Area code is first 3 digits)

```
#CODE HERE

print('Code - Total Count')
hotels['phone-number'].apply(lambda num:num[:3]).value_counts()[:3]
```

```
Code - Total Count

799    168
185    167
541    166
Name: phone-number, dtype: int64
```

TASK: How many arrivals took place between the 1st and the 15th of the month (inclusive of 1 and 15) ? Bonus: Can you do this in one line of pandas code?

```
#CODE HERE

hotels['arrival_date_day_of_month'].apply(lambda day: day in
range(1,16)).sum()

58152
```

HARD BONUS TASK: Create a table for counts for each day of the week that people arrived. (E.g. 5000 arrivals were on a Monday, 3000 were on a Tuesday, etc..)

```
# CODE HERE

import numpy as np

def convert(day,month,year):
    return f'{day}-{month}-{year}'

hotels['date']  = np.vectorize(convert)
(hotels['arrival_date_day_of_month'],
                                        hotels['arrival_date_month'],
                                        hotels['arrival_date_year'])

hotels['date'] = pd.to_datetime(hotels['date'])

#
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Seri
es.dt.day_name.html#pandas.Series.dt.day_name
hotels['date'].dt.day_name().value_counts()

Friday       19631
Thursday     19254
Monday       18171
Saturday     18055
Wednesday    16139
Sunday       14141
Tuesday      13999
Name: date, dtype: int64
```