

(Hotel Booking).Project Data Analysis

December 10, 2023

```
[5]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
[7]: df=pd.read_csv('hotel_bookings 2.csv')
```

```
[10]: df.head() #exploratory data analsis and data cleaning
```

```
[10]:
```

	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	...	deposit_type	\
0	0	0	2	...	No Deposit	
1	0	0	2	...	No Deposit	
2	0	1	1	...	No Deposit	
3	0	1	1	...	No Deposit	
4	0	2	2	...	No Deposit	

	agent	company	days_in_waiting_list	customer_type	adr	\
0	NaN	NaN	0	Transient	0.0	
1	NaN	NaN	0	Transient	0.0	
2	NaN	NaN	0	Transient	75.0	
3	304.0	NaN	0	Transient	75.0	
4	240.0	NaN	0	Transient	98.0	

	required_car_parking_spaces	total_of_special_requests	reservation_status	\
0	0	0	Check-Out	
1	0	0	Check-Out	
2	0	0	Check-Out	
3	0	0	Check-Out	
4	0	1	Check-Out	

	reservation_status_date
0	1/7/2015
1	1/7/2015
2	2/7/2015
3	2/7/2015
4	3/7/2015

[5 rows x 32 columns]

```
[16]: df.tail()
```

```
[16]:
```

	hotel	is_canceled	lead_time	arrival_date_year	\
119385	City Hotel	0	23	2017	
119386	City Hotel	0	102	2017	
119387	City Hotel	0	34	2017	
119388	City Hotel	0	109	2017	
119389	City Hotel	0	205	2017	

	arrival_date_month	arrival_date_week_number	\
119385	August	35	
119386	August	35	
119387	August	35	
119388	August	35	
119389	August	35	

	arrival_date_day_of_month	stays_in_weekend_nights	\
119385	30	2	
119386	31	2	
119387	31	2	
119388	31	2	
119389	29	2	

	stays_in_week_nights	adults	...	deposit_type	agent	company	\
119385	5	2	...	No Deposit	394.0	NaN	
119386	5	3	...	No Deposit	9.0	NaN	
119387	5	2	...	No Deposit	9.0	NaN	
119388	5	2	...	No Deposit	89.0	NaN	
119389	7	2	...	No Deposit	9.0	NaN	

	days_in_waiting_list	customer_type	adr \
119385	0	Transient	96.14
119386	0	Transient	225.43
119387	0	Transient	157.71
119388	0	Transient	104.40
119389	0	Transient	151.20

	required_car_parking_spaces	total_of_special_requests \
119385	0	0
119386	0	2
119387	0	4
119388	0	0
119389	0	2

	reservation_status	reservation_status_date
119385	Check-Out	6/9/2017
119386	Check-Out	7/9/2017
119387	Check-Out	7/9/2017
119388	Check-Out	7/9/2017
119389	Check-Out	7/9/2017

[5 rows x 32 columns]

```
[18]: df.shape
```

```
[18]: (119390, 32)
```

```
[20]: df.columns
```

```
[20]: Index(['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', 'children', 'babies', 'meal',
        'country', 'market_segment', 'distribution_channel',
        'is_repeated_guest', 'previous_cancellations',
        'previous_bookings_not_canceled', 'reserved_room_type',
        'assigned_room_type', 'booking_changes', 'deposit_type', 'agent',
        'company', 'days_in_waiting_list', 'customer_type', 'adr',
        'required_car_parking_spaces', 'total_of_special_requests',
        'reservation_status', 'reservation_status_date'],
        dtype='object')
```

```
[24]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 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

dtypes: float64(4), int64(16), object(12)

memory usage: 29.1+ MB

```
[26]: df['reservation_status_date'] = pd.to_datetime(df['reservation_status_date'])
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[26], line 1
----> 1 df['reservation_status_date'] = pd.
      ↪to_datetime(df['reservation_status_date'])

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
      ↪pandas/core/tools/datetimes.py:1046, in to_datetime(arg, errors, dayfirst,
      ↪yearfirst, utc, format, exact, unit, infer_datetime_format, origin, cache)
```

```

1044         result = arg.tz_localize("utc")
1045 elif isinstance(arg, ABCSeries):
-> 1046     cache_array = _maybe_cache(arg, format, cache, convert_listlike)
1047     if not cache_array.empty:
1048         result = arg.map(cache_array)

```

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
↳ pandas/core/tools/datetimes.py:250, in _maybe_cache(arg, format, cache,
↳ convert_listlike)

```

248 unique_dates = unique(arg)
249 if len(unique_dates) < len(arg):
--> 250     cache_dates = convert_listlike(unique_dates, format)
251     # GH#45319
252     try:

```

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
↳ pandas/core/tools/datetimes.py:453, in _convert_listlike_datetimes(arg,
↳ format, name, utc, unit, errors, dayfirst, yearfirst, exact)

```

451 # `format` could be inferred, or user didn't ask for mixed-format   

↳ parsing.
452 if format is not None and format != "mixed":
--> 453     return _array_strptime_with_fallback(arg, name, utc, format, exact,   

↳ errors)
455 result, tz_parsed = objects_to_datetime64ns(
456     arg,
457     dayfirst=dayfirst,
458     (...)
461     allow_object=True,
462 )
464 if tz_parsed is not None:
465     # We can take a shortcut since the datetime64 numpy array
466     # is in UTC

```

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
↳ pandas/core/tools/datetimes.py:484, in _array_strptime_with_fallback(arg,
↳ name, utc, fmt, exact, errors)

```

473 def _array_strptime_with_fallback(
474     arg,
475     name,
476     (...)
479     errors: str,
480 ) -> Index:
481     """
482     Call array_strptime, with fallback behavior depending on 'errors'.
483     """
--> 484     result, timezones = array_strptime(arg, fmt, exact=exact,   

↳ errors=errors, utc=utc)
485     if any(tz is not None for tz in timezones):

```

```

486         return _return_parsed_timezone_results(result, timezones, utc,
↳name)

```

```

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
↳pandas/_libs/tslibs/strptime.pyx:530, in pandas._libs.tslibs.strptime.
↳array_strptime()

```

```

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
↳pandas/_libs/tslibs/strptime.pyx:351, in pandas._libs.tslibs.strptime.
↳array_strptime()

```

```

ValueError: time data "22/4/2015" doesn't match format "%m/%d/%Y", at position .
↳ You might want to try:
    - passing `format` if your strings have a consistent format;
    - passing `format='ISO8601'` if your strings are all ISO8601 but not
↳necessarily in exactly the same format;
    - passing `format='mixed'`, and the format will be inferred for each elemen
↳individually. You might want to use `dayfirst` alongside this.

```

```
[28]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 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
dtypes: float64(4), int64(16), object(12)
memory usage: 29.1+ MB

```

```
[30]: df.describe(include = 'object')
```

```

[30]:
      hotel arrival_date_month  meal country market_segment \
count      119390           119390  119390  118902           119390
unique         2             12      5      177             8
top    City Hotel      August    BB    PRT      Online TA
freq      79330           13877   92310   48590           56477

      distribution_channel reserved_room_type assigned_room_type \
count           119390           119390           119390
unique             5             10             12
top           TA/TO             A             A
freq           97870           85994           74053

      deposit_type customer_type reservation_status reservation_status_date
count      119390           119390           119390           119390
unique         3             4             3             926
top    No Deposit    Transient    Check-Out           21/10/2015
freq      104641           89613           75166           1461

```

```

[34]: for col in df.describe(include = 'object').columns:
      print(col)
      print(df[col].unique())
      print('-'*50)

```

```

hotel
['Resort Hotel' 'City Hotel']
-----
arrival_date_month
['July' 'August' 'September' 'October' 'November' 'December' 'January'
 'February' 'March' 'April' 'May' 'June']
-----
meal
['BB' 'FB' 'HB' 'SC' 'Undefined']
-----

```

country

['PRT' 'GBR' 'USA' 'ESP' 'IRL' 'FRA' nan 'ROU' 'NOR' 'OMN' 'ARG' 'POL'
'DEU' 'BEL' 'CHE' 'CN' 'GRC' 'ITA' 'NLD' 'DNK' 'RUS' 'SWE' 'AUS' 'EST'
'CZE' 'BRA' 'FIN' 'MOZ' 'BWA' 'LUX' 'SVN' 'ALB' 'IND' 'CHN' 'MEX' 'MAR'
'UKR' 'SMR' 'LVA' 'PRI' 'SRB' 'CHL' 'AUT' 'BLR' 'LTU' 'TUR' 'ZAF' 'AGO'
'ISR' 'CYM' 'ZMB' 'CPV' 'ZWE' 'DZA' 'KOR' 'CRI' 'HUN' 'ARE' 'TUN' 'JAM'
'HRV' 'HKG' 'IRN' 'GEO' 'AND' 'GIB' 'URY' 'JEY' 'CAF' 'CYP' 'COL' 'GGY'
'KWT' 'NGA' 'MDV' 'VEN' 'SVK' 'FJI' 'KAZ' 'PAK' 'IDN' 'LBN' 'PHL' 'SEN'
'SYC' 'AZE' 'BHR' 'NZL' 'THA' 'DOM' 'MKD' 'MYS' 'ARM' 'JPN' 'LKA' 'CUB'
'CMR' 'BIH' 'MUS' 'COM' 'SUR' 'UGA' 'BGR' 'CIV' 'JOR' 'SYR' 'SGP' 'BDI'
'SAU' 'VNM' 'PLW' 'QAT' 'EGY' 'PER' 'MLT' 'MWI' 'ECU' 'MDG' 'ISL' 'UZB'
'NPL' 'BHS' 'MAC' 'TGO' 'TWN' 'DJI' 'STP' 'KNA' 'ETH' 'IRQ' 'HND' 'RWA'
'KHM' 'MCO' 'BGD' 'IMN' 'TJK' 'NIC' 'BEN' 'VGB' 'TZA' 'GAB' 'GHA' 'TMP'
'GLP' 'KEN' 'LIE' 'GNB' 'MNE' 'UMI' 'MYT' 'FRO' 'MMR' 'PAN' 'BFA' 'LBY'
'MLI' 'NAM' 'BOL' 'PRY' 'BRB' 'ABW' 'AIA' 'SLV' 'DMA' 'PYF' 'GUY' 'LCA'
'ATA' 'GTM' 'ASM' 'MRT' 'NCL' 'KIR' 'SDN' 'ATF' 'SLE' 'LAO']

market_segment

['Direct' 'Corporate' 'Online TA' 'Offline TA/TO' 'Complementary' 'Groups'
'Undefined' 'Aviation']

distribution_channel

['Direct' 'Corporate' 'TA/TO' 'Undefined' 'GDS']

reserved_room_type

['C' 'A' 'D' 'E' 'G' 'F' 'H' 'L' 'P' 'B']

assigned_room_type

['C' 'A' 'D' 'E' 'G' 'F' 'I' 'B' 'H' 'P' 'L' 'K']

deposit_type

['No Deposit' 'Refundable' 'Non Refund']

customer_type

['Transient' 'Contract' 'Transient-Party' 'Group']

reservation_status

['Check-Out' 'Canceled' 'No-Show']

reservation_status_date

['1/7/2015' '2/7/2015' '3/7/2015' '6/5/2015' '22/4/2015' '23/6/2015'
'5/7/2015' '6/7/2015' '7/7/2015' '8/7/2015' '11/5/2015' '15/7/2015'
'16/7/2015' '29/5/2015' '19/5/2015' '19/6/2015' '23/5/2015' '18/5/2015'
'9/7/2015' '2/6/2015' '13/7/2015' '4/7/2015' '29/6/2015' '16/6/2015'
'18/6/2015' '12/6/2015' '9/6/2015' '26/5/2015' '11/7/2015' '12/7/2015'
'17/7/2015' '15/4/2015' '13/5/2015' '10/7/2015' '20/5/2015' '12/5/2015'
'14/7/2015' '17/6/2015' '1/5/2015' '30/3/2015' '19/7/2015' '3/6/2015'
'26/6/2015' '14/5/2015' '20/7/2015' '7/5/2015' '28/5/2015' '13/4/2015']

'25/3/2015' '21/7/2015' '27/6/2015' '18/7/2015' '23/7/2015' '8/6/2015'
'22/6/2015' '24/6/2015' '5/3/2015' '1/6/2015' '24/4/2015' '22/7/2015'
'27/5/2015' '6/4/2015' '11/4/2015' '25/7/2015' '28/7/2015' '29/7/2015'
'25/6/2015' '24/7/2015' '5/6/2015' '30/6/2015' '13/6/2015' '11/6/2015'
'30/7/2015' '27/7/2015' '29/4/2015' '4/6/2015' '26/7/2015' '1/8/2015'
'2/8/2015' '15/6/2015' '23/4/2015' '31/7/2015' '25/5/2015' '3/8/2015'
'17/4/2015' '4/8/2015' '6/8/2015' '15/5/2015' '9/5/2015' '17/3/2015'
'22/5/2015' '7/8/2015' '4/4/2015' '5/8/2015' '8/8/2015' '10/8/2015'
'4/5/2015' '6/6/2015' '9/8/2015' '15/8/2015' '11/8/2015' '28/3/2015'
'14/8/2015' '12/8/2015' '16/8/2015' '16/5/2015' '21/8/2015' '13/8/2015'
'17/8/2015' '20/4/2015' '18/8/2015' '23/8/2015' '22/8/2015' '19/8/2015'
'20/8/2015' '29/8/2015' '31/3/2015' '30/5/2015' '25/8/2015' '14/4/2015'
'24/8/2015' '24/3/2015' '21/5/2015' '28/8/2015' '26/8/2015' '27/8/2015'
'30/8/2015' '31/8/2015' '6/9/2015' '3/9/2015' '4/9/2015' '2/9/2015'
'1/9/2015' '5/9/2015' '20/6/2015' '7/9/2015' '10/9/2015' '11/9/2015'
'8/9/2015' '9/9/2015' '13/9/2015' '15/9/2015' '10/4/2015' '2/1/2015'
'18/11/2014' '12/9/2015' '17/9/2015' '14/9/2015' '7/4/2015' '19/9/2015'
'16/9/2015' '20/9/2015' '18/1/2015' '23/10/2015' '22/1/2015' '1/1/2015'
'22/9/2015' '24/9/2015' '18/9/2015' '21/9/2015' '30/9/2015' '25/9/2015'
'27/9/2015' '28/9/2015' '12/10/2015' '29/9/2015' '23/9/2015' '1/10/2015'
'26/9/2015' '18/4/2015' '2/10/2015' '4/10/2015' '8/10/2015' '3/10/2015'
'7/10/2015' '9/10/2015' '11/10/2015' '5/10/2015' '6/10/2015' '10/10/2015'
'14/10/2015' '15/10/2015' '18/10/2015' '13/10/2015' '20/10/2015'
'19/10/2015' '31/10/2015' '16/10/2015' '21/10/2015' '22/10/2015'
'17/10/2015' '24/10/2015' '25/10/2015' '28/10/2015' '27/10/2015'
'26/10/2015' '30/10/2015' '5/11/2015' '29/10/2015' '3/11/2015'
'7/11/2015' '4/11/2015' '1/11/2015' '2/11/2015' '17/11/2015' '6/11/2015'
'10/11/2015' '8/11/2015' '9/11/2015' '15/11/2015' '16/11/2015'
'11/11/2015' '12/11/2015' '14/11/2015' '13/11/2015' '18/11/2015'
'22/11/2015' '19/11/2015' '21/11/2015' '20/11/2015' '24/11/2015'
'25/11/2015' '23/11/2015' '28/11/2015' '26/11/2015' '27/11/2015'
'29/11/2015' '4/12/2015' '1/12/2015' '6/12/2015' '8/12/2015' '2/12/2015'
'3/12/2015' '31/12/2015' '5/12/2015' '10/12/2015' '17/12/2015'
'30/11/2015' '12/12/2015' '7/12/2015' '5/1/2016' '11/12/2015'
'13/12/2015' '15/12/2015' '16/12/2015' '19/12/2015' '18/12/2015'
'26/12/2015' '27/12/2015' '22/12/2015' '23/12/2015' '24/12/2015'
'29/12/2015' '28/12/2015' '20/12/2015' '30/12/2015' '2/1/2016' '1/1/2016'
'25/12/2015' '3/1/2016' '4/1/2016' '11/1/2016' '7/1/2016' '21/12/2015'
'9/1/2016' '10/1/2016' '8/1/2016' '6/1/2016' '12/1/2016' '13/1/2016'
'23/1/2016' '9/2/2016' '15/1/2016' '16/1/2016' '17/1/2016' '19/1/2016'
'18/1/2016' '21/1/2016' '24/1/2016' '22/1/2016' '29/1/2016' '27/1/2016'
'25/1/2016' '8/3/2016' '26/1/2016' '20/1/2016' '30/1/2016' '1/2/2016'
'2/2/2016' '8/2/2016' '7/2/2016' '28/1/2016' '5/2/2016' '3/2/2016'
'13/2/2016' '10/2/2016' '4/2/2016' '12/2/2016' '11/2/2016' '16/2/2016'
'14/2/2016' '15/2/2016' '20/2/2016' '6/2/2016' '14/1/2016' '17/2/2016'
'21/2/2016' '24/2/2016' '25/2/2016' '19/2/2016' '18/2/2016' '26/2/2016'
'23/2/2016' '5/3/2016' '22/2/2016' '27/2/2016' '3/3/2016' '24/3/2016'
'4/3/2016' '29/2/2016' '1/3/2016' '2/3/2016' '30/3/2016' '7/3/2016'

'14/3/2016' '21/3/2016' '9/3/2016' '12/3/2016' '22/3/2016' '10/3/2016'
'11/3/2016' '20/3/2016' '15/3/2016' '17/3/2016' '16/3/2016' '19/3/2016'
'27/3/2016' '18/3/2016' '26/3/2016' '31/3/2016' '28/3/2016' '29/3/2016'
'1/4/2016' '23/3/2016' '2/4/2016' '25/3/2016' '13/3/2016' '4/4/2016'
'3/4/2016' '5/4/2016' '8/4/2016' '6/4/2016' '9/4/2016' '12/4/2016'
'16/4/2016' '17/4/2016' '27/4/2016' '14/4/2016' '18/4/2016' '21/4/2016'
'19/4/2016' '20/4/2016' '10/4/2016' '13/4/2016' '11/4/2016' '7/4/2016'
'15/4/2016' '22/4/2016' '23/4/2016' '26/4/2016' '28/4/2016' '24/4/2016'
'25/4/2016' '29/4/2016' '30/4/2016' '1/5/2016' '10/5/2016' '2/5/2016'
'7/5/2016' '8/5/2016' '12/5/2016' '4/5/2016' '6/5/2016' '3/5/2016'
'9/5/2016' '5/5/2016' '13/5/2016' '14/5/2016' '18/5/2016' '19/5/2016'
'15/5/2016' '16/5/2016' '11/5/2016' '21/5/2016' '22/5/2016' '20/5/2016'
'24/5/2016' '25/5/2016' '26/5/2016' '23/5/2016' '27/5/2016' '17/5/2016'
'29/5/2016' '28/5/2016' '30/5/2016' '31/5/2016' '1/6/2016' '3/6/2016'
'8/6/2016' '2/6/2016' '5/6/2016' '6/6/2016' '13/6/2016' '7/6/2016'
'10/6/2016' '11/6/2016' '16/6/2016' '12/6/2016' '14/6/2016' '17/6/2016'
'4/6/2016' '18/6/2016' '21/6/2016' '9/6/2016' '24/6/2016' '20/6/2016'
'25/6/2016' '22/6/2016' '26/6/2016' '23/6/2016' '1/7/2016' '15/6/2016'
'28/6/2016' '2/7/2016' '19/6/2016' '27/6/2016' '4/7/2016' '30/6/2016'
'5/7/2016' '8/7/2016' '9/7/2016' '7/7/2016' '12/7/2016' '29/6/2016'
'10/7/2016' '15/7/2016' '3/7/2016' '16/7/2016' '14/7/2016' '18/7/2016'
'13/7/2016' '6/7/2016' '20/7/2016' '21/7/2016' '23/7/2016' '19/7/2016'
'11/7/2016' '28/7/2016' '17/7/2016' '25/7/2016' '22/7/2016' '29/7/2016'
'3/8/2016' '2/8/2016' '4/8/2016' '8/8/2016' '10/8/2016' '1/8/2016'
'6/8/2016' '6/3/2016' '5/8/2016' '26/7/2016' '7/8/2016' '30/7/2016'
'24/7/2016' '12/8/2016' '27/7/2016' '13/8/2016' '18/8/2016' '16/8/2016'
'15/8/2016' '17/8/2016' '11/8/2016' '31/7/2016' '19/8/2016' '1/9/2016'
'23/8/2016' '26/8/2016' '20/8/2016' '21/8/2016' '4/9/2016' '22/8/2016'
'27/8/2016' '25/8/2016' '9/8/2016' '5/9/2016' '24/8/2016' '10/9/2016'
'29/8/2016' '9/9/2016' '30/8/2016' '13/9/2016' '31/8/2016' '14/9/2016'
'12/9/2016' '15/9/2016' '14/8/2016' '2/9/2016' '8/9/2016' '19/9/2016'
'16/9/2016' '7/9/2016' '21/9/2016' '6/9/2016' '22/9/2016' '17/9/2016'
'20/9/2016' '3/9/2016' '26/9/2016' '23/9/2016' '18/9/2016' '29/9/2016'
'2/10/2016' '1/10/2016' '27/9/2016' '25/9/2016' '5/10/2016' '11/9/2016'
'30/9/2016' '9/10/2016' '3/10/2016' '6/10/2016' '11/10/2016' '24/9/2016'
'13/10/2016' '28/9/2016' '8/10/2016' '7/10/2016' '16/10/2016' '28/8/2016'
'17/10/2016' '18/10/2016' '10/10/2016' '4/10/2016' '15/10/2016'
'19/10/2016' '21/10/2016' '12/10/2016' '24/10/2016' '26/10/2016'
'23/10/2016' '20/10/2016' '25/10/2016' '27/10/2016' '28/10/2016'
'30/10/2016' '29/10/2016' '1/11/2016' '4/11/2016' '14/10/2016'
'7/11/2016' '3/11/2016' '10/11/2016' '14/11/2016' '2/11/2016'
'31/10/2016' '11/11/2016' '8/11/2016' '5/11/2016' '25/11/2016'
'9/11/2016' '20/11/2016' '21/11/2016' '22/10/2016' '22/11/2016'
'16/11/2016' '23/11/2016' '17/11/2016' '6/11/2016' '15/11/2016'
'13/11/2016' '12/11/2016' '27/11/2016' '19/11/2016' '30/11/2016'
'18/11/2016' '2/12/2016' '4/12/2016' '29/11/2016' '7/12/2016'
'28/11/2016' '3/12/2016' '6/12/2016' '24/11/2016' '8/12/2016' '5/12/2016'
'10/12/2016' '13/12/2016' '14/12/2016' '16/12/2016' '15/12/2016'

'17/12/2016' '19/12/2016' '21/12/2016' '20/12/2016' '22/12/2016'
 '23/12/2016' '24/12/2016' '1/12/2016' '27/12/2016' '29/12/2016'
 '30/12/2016' '12/12/2016' '2/1/2017' '11/12/2016' '3/1/2017' '4/1/2017'
 '1/1/2017' '26/12/2016' '6/1/2017' '28/12/2016' '18/12/2016' '10/1/2017'
 '11/1/2017' '7/1/2017' '12/1/2017' '16/1/2017' '14/1/2017' '13/1/2017'
 '5/1/2017' '17/1/2017' '20/1/2017' '9/12/2016' '26/1/2017' '31/12/2016'
 '23/1/2017' '27/1/2017' '28/1/2017' '19/1/2017' '25/1/2017' '24/1/2017'
 '29/1/2017' '18/1/2017' '25/12/2016' '15/1/2017' '21/1/2017' '1/2/2017'
 '2/2/2017' '31/1/2017' '3/2/2017' '4/2/2017' '6/2/2017' '7/2/2017'
 '8/2/2017' '30/1/2017' '9/2/2017' '9/1/2017' '11/2/2017' '10/2/2017'
 '12/2/2017' '13/2/2017' '14/2/2017' '16/2/2017' '17/2/2017' '18/2/2017'
 '19/2/2017' '20/2/2017' '15/2/2017' '21/2/2017' '22/2/2017' '26/2/2017'
 '23/2/2017' '24/2/2017' '25/2/2017' '28/2/2017' '5/3/2017' '27/2/2017'
 '3/3/2017' '6/3/2017' '2/3/2017' '8/3/2017' '9/3/2017' '10/3/2017'
 '7/3/2017' '12/3/2017' '13/3/2017' '14/3/2017' '1/3/2017' '18/3/2017'
 '17/3/2017' '24/3/2017' '22/3/2017' '26/3/2017' '27/3/2017' '11/3/2017'
 '28/3/2017' '29/3/2017' '30/3/2017' '31/3/2017' '19/3/2017' '22/1/2017'
 '2/4/2017' '20/3/2017' '3/4/2017' '8/1/2017' '23/3/2017' '5/4/2017'
 '5/2/2017' '4/4/2017' '15/3/2017' '7/4/2017' '25/3/2017' '8/4/2017'
 '6/4/2017' '21/3/2017' '10/4/2017' '1/4/2017' '11/4/2017' '13/4/2017'
 '15/4/2017' '12/4/2017' '4/3/2017' '19/4/2017' '22/4/2017' '20/4/2017'
 '2/5/2017' '9/4/2017' '23/4/2017' '24/4/2017' '16/4/2017' '28/4/2017'
 '18/4/2017' '26/4/2017' '25/4/2017' '17/4/2017' '21/4/2017' '3/5/2017'
 '4/5/2017' '16/3/2017' '5/5/2017' '29/4/2017' '14/4/2017' '8/5/2017'
 '27/4/2017' '11/5/2017' '1/5/2017' '10/5/2017' '13/5/2017' '6/5/2017'
 '14/5/2017' '16/5/2017' '30/4/2017' '15/5/2017' '7/5/2017' '9/5/2017'
 '17/5/2017' '21/5/2017' '12/5/2017' '22/5/2017' '24/5/2017' '23/5/2017'
 '25/5/2017' '26/5/2017' '28/5/2017' '27/5/2017' '29/5/2017' '19/5/2017'
 '31/5/2017' '20/5/2017' '1/6/2017' '30/5/2017' '2/6/2017' '26/11/2016'
 '4/6/2017' '5/6/2017' '6/6/2017' '7/6/2017' '18/5/2017' '9/6/2017'
 '10/6/2017' '11/6/2017' '12/6/2017' '14/6/2017' '8/6/2017' '16/6/2017'
 '13/6/2017' '3/6/2017' '24/6/2017' '20/6/2017' '19/6/2017' '21/6/2017'
 '26/6/2017' '27/6/2017' '22/6/2017' '28/6/2017' '15/6/2017' '29/6/2017'
 '30/6/2017' '18/6/2017' '4/7/2017' '8/7/2017' '5/7/2017' '3/7/2017'
 '7/7/2017' '1/7/2017' '6/7/2017' '11/7/2017' '12/7/2017' '23/6/2017'
 '13/7/2017' '2/7/2017' '10/7/2017' '14/7/2017' '15/7/2017' '16/7/2017'
 '18/7/2017' '17/7/2017' '19/7/2017' '20/7/2017' '21/7/2017' '25/6/2017'
 '17/6/2017' '24/7/2017' '26/7/2017' '9/7/2017' '27/7/2017' '28/7/2017'
 '31/7/2017' '29/7/2017' '22/7/2017' '2/8/2017' '1/8/2017' '3/8/2017'
 '4/8/2017' '25/7/2017' '23/7/2017' '9/8/2017' '10/8/2017' '30/7/2017'
 '7/8/2017' '13/8/2017' '5/8/2017' '14/8/2017' '8/8/2017' '16/8/2017'
 '17/8/2017' '15/8/2017' '18/8/2017' '20/8/2017' '22/8/2017' '6/8/2017'
 '25/8/2017' '26/8/2017' '23/8/2017' '11/8/2017' '27/8/2017' '21/8/2017'
 '29/8/2017' '31/8/2017' '12/8/2017' '19/8/2017' '31/1/2016' '1/9/2017'
 '28/8/2017' '3/4/2015' '21/1/2015' '28/1/2015' '29/1/2015' '30/1/2015'
 '2/2/2015' '5/2/2015' '6/2/2015' '9/2/2015' '10/2/2015' '11/2/2015'
 '12/2/2015' '19/2/2015' '20/2/2015' '23/2/2015' '24/2/2015' '25/2/2015'
 '26/2/2015' '27/2/2015' '3/3/2015' '4/3/2015' '6/3/2015' '9/3/2015'

```
'11/3/2015' '12/3/2015' '18/3/2015' '2/4/2015' '14/6/2015' '8/4/2015'
'16/4/2015' '25/4/2015' '28/4/2015' '8/5/2015' '6/9/2017' '28/2/2016'
'9/12/2015' '14/12/2015' '9/9/2017' '2/9/2017' '24/8/2017' '30/8/2017'
'3/9/2017' '4/9/2017' '5/9/2017' '7/9/2017' '8/9/2017' '10/9/2017'
'12/9/2017' '14/9/2017' '30/4/2015' '21/4/2015' '5/4/2015' '13/3/2015'
'5/5/2015' '29/3/2015' '10/6/2015' '27/4/2015' '17/10/2014' '20/1/2015'
'17/2/2015' '10/3/2015' '23/3/2015']
```

```
[36]: df.isnull().sum()
```

```
[36]: 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
      dtype: int64
```

```
[38]: df.drop(['company', 'agent'], axis = 1, inplace = True)
      df.dropna(inplace = True)
```

```
[42]: df.isnull().sum()
```

```
[42]: 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            0
      babies              0
      meal                0
      country             0
      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
      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
      dtype: int64
```

```
[44]: df.describe()
```

```
[44]:
```

	is_canceled	lead_time	arrival_date_year	\
count	118898.000000	118898.000000	118898.000000	
mean	0.371352	104.311435	2016.157656	
std	0.483168	106.903309	0.707459	
min	0.000000	0.000000	2015.000000	
25%	0.000000	18.000000	2016.000000	
50%	0.000000	69.000000	2016.000000	
75%	1.000000	161.000000	2017.000000	
max	1.000000	737.000000	2017.000000	

	arrival_date_week_number	arrival_date_day_of_month	\
--	--------------------------	---------------------------	---

count	118898.000000	118898.000000
mean	27.166555	15.800880
std	13.589971	8.780324
min	1.000000	1.000000
25%	16.000000	8.000000
50%	28.000000	16.000000
75%	38.000000	23.000000
max	53.000000	31.000000

	stays_in_weekend_nights	stays_in_week_nights	adults \
count	118898.000000	118898.000000	118898.000000
mean	0.928897	2.502145	1.858391
std	0.996216	1.900168	0.578576
min	0.000000	0.000000	0.000000
25%	0.000000	1.000000	2.000000
50%	1.000000	2.000000	2.000000
75%	2.000000	3.000000	2.000000
max	16.000000	41.000000	55.000000

	children	babies	is_repeated_guest \
count	118898.000000	118898.000000	118898.000000
mean	0.104207	0.007948	0.032011
std	0.399172	0.097380	0.176029
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000
max	10.000000	10.000000	1.000000

	previous_cancellations	previous_bookings_not_canceled \
count	118898.000000	118898.000000
mean	0.087142	0.131634
std	0.845869	1.484672
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	0.000000
75%	0.000000	0.000000
max	26.000000	72.000000

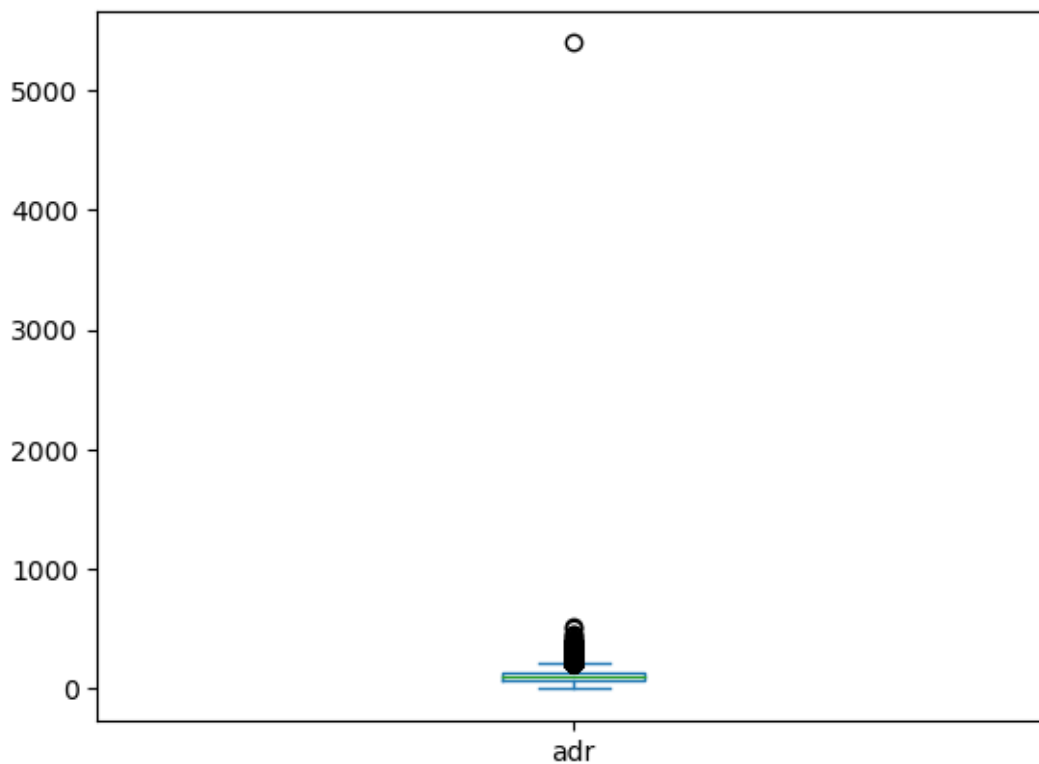
	booking_changes	days_in_waiting_list	adr \
count	118898.000000	118898.000000	118898.000000
mean	0.221181	2.330754	102.003243
std	0.652785	17.630452	50.485862
min	0.000000	0.000000	-6.380000
25%	0.000000	0.000000	70.000000
50%	0.000000	0.000000	95.000000
75%	0.000000	0.000000	126.000000

max	21.000000	391.000000	5400.000000
-----	-----------	------------	-------------

	required_car_parking_spaces	total_of_special_requests
count	118898.000000	118898.000000
mean	0.061885	0.571683
std	0.244172	0.792678
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	0.000000
75%	0.000000	1.000000
max	8.000000	5.000000

```
[46]: df['adr'].plot(kind = 'box')
```

```
[46]: <Axes: >
```



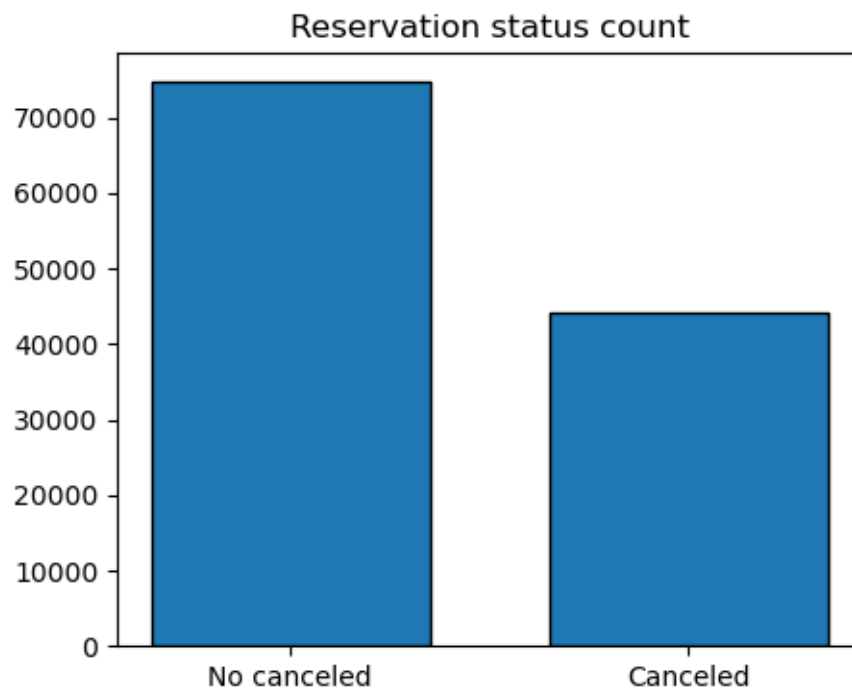
```
[48]: df = df[df['adr']<5000]
```

```
[52]: #Data Analysis & Visualizations
```

```
cancelled_perc = df['is_cancelled'].value_counts(normalize = True)
print(cancelled_perc)
```

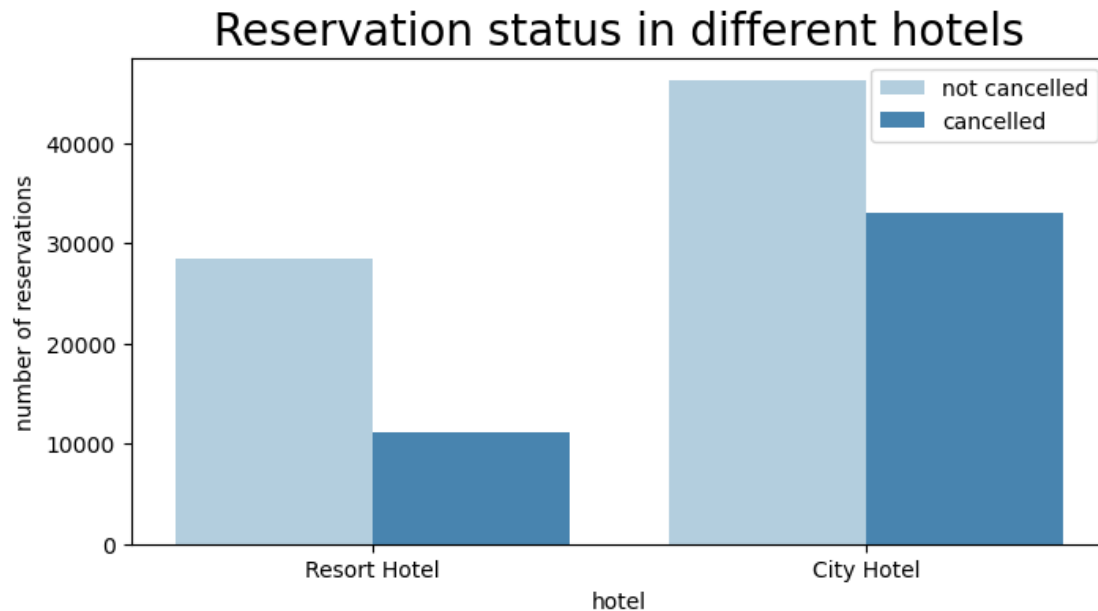
```
plt.figure(figsize = (5,4))
plt.title('Reservation status count')
plt.bar(['No canceled','Canceled'],df['is_canceled'].value_counts(), edgecolor='k',width=0.7)
plt.show()
```

```
is_canceled
0    0.628653
1    0.371347
Name: proportion, dtype: float64
```



```
[58]: plt.figure(figsize = (8,4))
axl=sns.countplot(x='hotel',hue='is_canceled',data=df,palette = 'Blues')
plt.title('Reservation status in different hotels',size = 20)
plt.xlabel('hotel')
plt.ylabel('number of reservations')
plt.legend(['not cancelled','cancelled'])
plt.plot()
```

[58]: []



```
[62]: resort_hotel = df[df['hotel'] == 'Resort Hotel']
      resort_hotel['is_canceled'].value_counts(normalize = True)
```

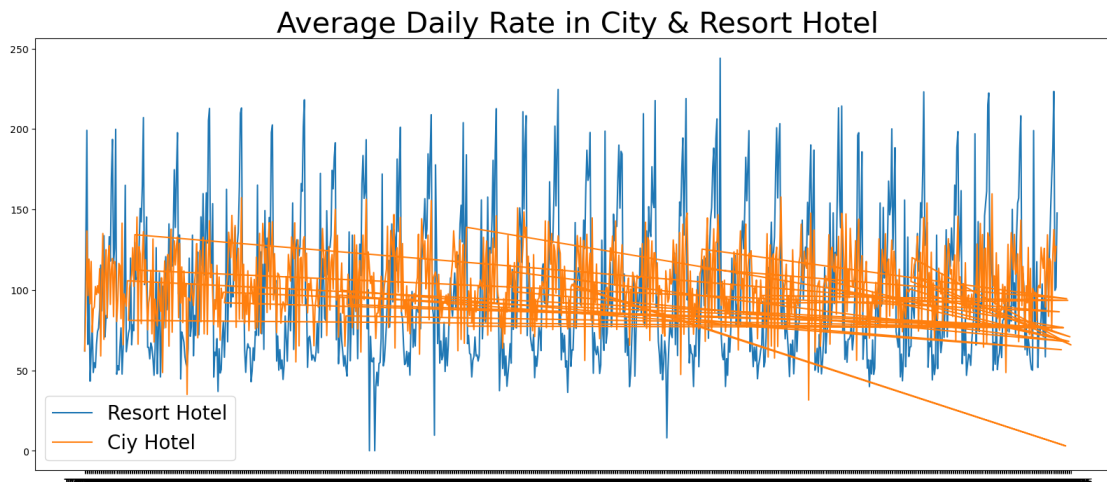
```
[62]: is_canceled
0      0.72025
1      0.27975
Name: proportion, dtype: float64
```

```
[64]: city_hotel = df[df['hotel'] == 'City Hotel']
      city_hotel['is_canceled'].value_counts(normalize = True)
```

```
[64]: is_canceled
0      0.582918
1      0.417082
Name: proportion, dtype: float64
```

```
[66]: resort_hotel = resort_hotel.groupby('reservation_status_date')[['adr']].mean()
      city_hotel = city_hotel.groupby('reservation_status_date')[['adr']].mean()
```

```
[72]: plt.figure(figsize = (20,8))
      plt.title('Average Daily Rate in City & Resort Hotel',fontsize = 30)
      plt.plot(resort_hotel.index,resort_hotel['adr'],label = 'Resort Hotel')
      plt.plot(city_hotel.index,city_hotel['adr'],label = 'Ciy Hotel')
      plt.legend(fontsize = 20)
      plt.show()
```



```
[88]: df['month'] = df['reservation_status_date'].dt.month
plt.figure(figsize=(16,8))
ax1=sns.countplot(x = 'month',hue = 'is_canceled',data = df,palette='bright')
legends_labels,_ = ax1. get_legend_handles_labels()
ax1.legend(bbox_to_anchor=(1,1))
plt.title('Reservation status per month',size=20)
plt.xlabel('month')
plt.ylabel('number of reservations')
plt.legend(['not_canceled','canceled'])
plt.show()
```

```
-----
AttributeError                                Traceback (most recent call last)
Cell In[88], line 1
----> 1 df['month'] = df['reservation_status_date'].dt.month
      2 plt.figure(figsize=(16,8))
      3 ax1=sns.countplot(x = 'month',hue = 'is_canceled',data = _
    ↪df,palette='bright')

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
    ↪pandas/core/generic.py:5989, in NDFrame.__getattr__(self, name)
    5982 if (
    5983     name not in self._internal_names_set
    5984     and name not in self._metadata
    5985     and name not in self._accessors
    5986     and self._info_axis._can_hold_identifiers_and_holds_name(name)
    5987 ):
    5988     return self[name]
-> 5989 return object.__getattr__(self, name)
```

```

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
↳ pandas/core/accessor.py:224, in CachedAccessor.__get__(self, obj, cls)
    221 if obj is None:
    222     # we're accessing the attribute of the class, i.e., Dataset.geo
    223     return self._accessor
--> 224 accessor_obj = self._accessor(obj)
    225 # Replace the property with the accessor object. Inspired by:
    226 # https://www.pydanny.com/cached-property.html
    227 # We need to use object.__setattr__ because we overwrite __setattr__ on
    228 # NDFrame
    229 object.__setattr__(obj, self._name, accessor_obj)

```

```

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
↳ pandas/core/indexes/accessors.py:580, in CombinedDatetimelikeProperties.
↳ __new__(cls, data)
    577 elif is_period_dtype(data.dtype):
    578     return PeriodProperties(data, orig)
--> 580 raise AttributeError("Can only use .dt accessor with datetimelike_
↳ values")

```

AttributeError: Can only use .dt accessor with datetimelike values

[]: