In [1]:
```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime as datetime

from scipy import stats
from datetime import datetime, timedelta

import warnings
warnings.filterwarnings('ignore')
pd.set_option('display.max_columns', None)
```

In [2]:
```
df_bookings = pd.read_csv('dubai_booking_data.csv')
df_events = pd.read_csv('events.csv')
```

## Preprocessing:

In [4]:
```
df_bookings.drop_duplicates(inplace=True)
df_bookings.head()
```

Out[4]:

| | booking_id | created_at | city | country | category_name | p |
|---|---|---|---|---|---|---|
| **0** | bid_+1fN3xKE6A7bLb3Mb8EHug== | 2021-01-01 03:43:45 UTC | Dubai | United Arab Emirates | Dubai Dhow Cruises | |
| **1** | bid_oeNEooQN+xls8LMa8z7sMg== | 2021-01-01 05:34:28 UTC | Dubai | United Arab Emirates | Dubai Dhow Cruises | |
| **2** | bid_+1E+CdL+uNH/9LMgCTphtw== | 2021-01-01 06:06:02 UTC | Dubai | United Arab Emirates | Ferrari World Tickets | |
| **3** | bid_o55xFpl5f2ndjyBjWqgq/g== | 2021-01-01 06:26:37 UTC | Dubai | United Arab Emirates | Museum of Illusions Tickets | |
| **4** | bid_Qj1OEgYvKdCD1YXJzq2+PQ== | 2021-01-01 06:45:53 UTC | Dubai | United Arab Emirates | Museum of Illusions Tickets | |

In [5]:
```
df_events.drop_duplicates(inplace=True)
df_events.head()
```

Out[5]:

| | event_id | event_timestamp | event_name | |
|---|---|---|---|---|
| **0** | evt_347e97a21c984d3790e0a52ebf0c2f99 | 2022-09-24 07:40:50 UTC | product_page_view | cı |
| **1** | evt_19f165ad9c1b4943b037652bce725072 | 2022-09-24 08:06:01 UTC | product_page_view | cı |
| **2** | evt_6daf55c2a5f24b0a9f058f35ac244e5b | 2022-12-24 05:35:26 UTC | product_page_view | |
| **3** | evt_77e141a5899a47caa4201a5da17a7d58 | 2022-09-25 12:58:14 UTC | product_page_view | |
| **4** | evt_8e5b21f606f1499f9cbace8f2141681c | 2022-09-25 13:13:14 UTC | product_page_view | |

In [6]:
```
df_bookings.shape, df_events.shape
```

Out[6]:
```
((457620, 18), (267717, 29))
```

## Booking data has more rows than events. That shouldn't be the case in an ideal scenario

In [8]:
```
df_bookings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 457620 entries, 0 to 457619
Data columns (total 18 columns):
 #   Column                            Non-Null Count   Dtype 
---  ------                            --------------   ----- 
 0   booking_id                        457620 non-null  object
 1   created_at                        457620 non-null  object
 2   city                              457620 non-null  object
 3   country                           457620 non-null  object
 4   category_name                     457620 non-null  object
 5   product_id                        457620 non-null  int64 
 6   product_name                      457620 non-null  object
 7   experience_date                   457620 non-null  object
 8   experience_time                   457620 non-null  object
 9   customer_id                       457620 non-null  object
 10  number_of_guests                  457620 non-null  int64 
 11  customer_country                  457620 non-null  object
 12  device                            457620 non-null  object
 13  is_logged_in                      457620 non-null  bool  
 14  web_session_traffic_source        405571 non-null  object
 15  web_session_traffic_medium        405571 non-null  object
 16  web_session_campaign_name         299317 non-null  object
 17  web_session_traffic_origin_country 405418 non-null  object
dtypes: bool(1), int64(2), object(15)
memory usage: 59.8+ MB
```

In [9]:
```
df_events.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 267717 entries, 0 to 267716
Data columns (total 29 columns):
 #   Column                             Non-Null Count   Dtype  
---  ------                             --------------   -----  
 0   event_id                           267717 non-null  object 
 1   event_timestamp                    267717 non-null  object 
 2   event_name                         267717 non-null  object 
 3   customer_id                        267717 non-null  object 
 4   session_id                         267717 non-null  object 
 5   product_id                         267717 non-null  int64  
 6   booking_id                         71703 non-null   object 
 7   event_type                         267717 non-null  object 
 8   position                           267717 non-null  object 
 9   page_type                          267717 non-null  object 
 10  device                             267717 non-null  object 
 11  is_logged_in                       267717 non-null  bool   
 12  smart_rec_enabled                  267717 non-null  int64  
 13  rec_impressions                    267717 non-null  int64  
 14  rec_clicks                         267717 non-null  int64  
 15  rec_slot_position                  29094 non-null   float64
 16  search_filters_used                267717 non-null  int64  
 17  num_filters_applied                267717 non-null  int64  
 18  search_query_length                267717 non-null  int64  
 19  web_session_traffic_origin_country 236662 non-null  object 
 20  web_session_traffic_source         236770 non-null  object 
 21  web_session_traffic_medium         236770 non-null  object 
 22  web_session_campaign_name          164538 non-null  object 
 23  city                               267717 non-null  object 
 24  is_first_time_customer             267717 non-null  int64  
 25  customer_lifetime_bookings         267717 non-null  int64  
 26  experiment_bucket                  267717 non-null  object 
 27  active_discount_flag               267717 non-null  int64  
 28  supply_tier                        267717 non-null  object 
dtypes: bool(1), float64(1), int64(10), object(17)
memory usage: 57.4+ MB
```

In [10]:
```python
print(df_bookings['created_at'].min(), df_bookings['created_at'].max())
print(df_events['event_timestamp'].min(), df_events['event_timestamp'].ma
```

```
2021-01-01 00:33:48 UTC 2022-12-31 23:51:43 UTC
2022-07-01 01:34:37 UTC 2022-12-31 23:47:35 UTC
```

## Event Data has 6 months of less data than booking. thats why events data has lesser rows.

In [12]:
```python
print(f"timezones in booking data: {df_bookings['created_at'].astype(str)
print(f"timezones in events data: {df_events['event_timestamp'].astype(st
```

```
timezones in booking data: ['UTC']
timezones in events data: ['UTC']
```

In [13]:
```python
df_bookings.dtypes
```

Out[13]:
```
booking_id                            object
created_at                            object
city                                  object
country                               object
category_name                         object
product_id                             int64
product_name                          object
experience_date                       object
experience_time                       object
customer_id                           object
number_of_guests                       int64
customer_country                      object
device                                object
is_logged_in                            bool
web_session_traffic_source            object
web_session_traffic_medium            object
web_session_campaign_name             object
web_session_traffic_origin_country    object
dtype: object
```

In [14]:
```
df_events.dtypes
```

Out[14]:
```
event_id                               object
event_timestamp                        object
event_name                             object
customer_id                            object
session_id                             object
product_id                              int64
booking_id                             object
event_type                             object
position                               object
page_type                              object
device                                 object
is_logged_in                             bool
smart_rec_enabled                       int64
rec_impressions                         int64
rec_clicks                              int64
rec_slot_position                     float64
search_filters_used                     int64
num_filters_applied                     int64
search_query_length                     int64
web_session_traffic_origin_country     object
web_session_traffic_source             object
web_session_traffic_medium             object
web_session_campaign_name              object
city                                   object
is_first_time_customer                  int64
customer_lifetime_bookings              int64
experiment_bucket                      object
active_discount_flag                    int64
supply_tier                            object
dtype: object
```

## Date/Time Conversion:

In [16]:
```
df_bookings['created_at'] = pd.to_datetime(df_bookings['created_at'], utc
df_bookings['experience_date'] = pd.to_datetime(df_bookings['experience_d
df_events['event_timestamp'] = pd.to_datetime(df_events['event_timestamp'
df_bookings['created_date'] = df_bookings['created_at'].dt.strftime('%Y%m
df_bookings['exp_date'] = df_bookings['experience_date'].dt.strftime('%Y%
df_events['event_date'] = df_events['event_timestamp'].dt.strftime('%Y%m%

df_bookings['created_time'] = df_bookings['created_at'].dt.strftime('%H%M
df_events['event_time'] = df_events['event_timestamp'].dt.strftime('%H%M%

df_bookings['created_hour'] = df_bookings['created_at'].dt.hour
df_events['event_hour'] = df_events['event_timestamp'].dt.hour

temp_timedelta = pd.to_timedelta(df_bookings['experience_time'], errors='
df_bookings['exp_time'] = (pd.to_datetime(0) + temp_timedelta).dt.strftim
df_bookings['exp_hour'] = temp_timedelta.dt.components.hours

df_bookings[['created_at', 'created_date', 'created_time', 'created_hour'
```

Out[16]:

| | created_at | created_date | created_time | created_hour | experience_date | ex |
|---|---|---|---|---|---|---|
| **0** | 2021-01-01 03:43:45+00:00 | 20210101 | 034345 | 3 | 2021-01-01 00:00:00+00:00 | 20 |
| **1** | 2021-01-01 05:34:28+00:00 | 20210101 | 053428 | 5 | 2021-01-01 00:00:00+00:00 | 20 |
| **2** | 2021-01-01 06:06:02+00:00 | 20210101 | 060602 | 6 | 2021-01-01 00:00:00+00:00 | 20 |
| **3** | 2021-01-01 06:26:37+00:00 | 20210101 | 062637 | 6 | 2021-01-01 00:00:00+00:00 | 20 |
| **4** | 2021-01-01 06:45:53+00:00 | 20210101 | 064553 | 6 | 2021-01-01 00:00:00+00:00 | 20 |

In [17]:
```
df_bookings[['created_at', 'created_date', 'created_time', 'created_hour'
```

Out[17]:
```
created_at         datetime64[ns, UTC]
created_date                    object
created_time                    object
created_hour                     int32
experience_date    datetime64[ns, UTC]
exp_date                        object
experience_time                 object
exp_time                        object
exp_hour                         int64
dtype: object
```

In [18]:
```
df_events[['event_timestamp', 'event_date', 'event_time', 'event_hour']].
```

Out[18]:

| | event_timestamp | event_date | event_time | event_hour |
|---|---|---|---|---|
| **0** | 2022-09-24 07:40:50+00:00 | 20220924 | 074050 | 7 |
| **1** | 2022-09-24 08:06:01+00:00 | 20220924 | 080601 | 8 |
| **2** | 2022-12-24 05:35:26+00:00 | 20221224 | 053526 | 5 |
| **3** | 2022-09-25 12:58:14+00:00 | 20220925 | 125814 | 12 |
| **4** | 2022-09-25 13:13:14+00:00 | 20220925 | 131314 | 13 |

In [19]:
```
df_events[['event_timestamp', 'event_date', 'event_time', 'event_hour']].
```

Out[19]:
```
event_timestamp    datetime64[ns, UTC]
event_date                      object
event_time                      object
event_hour                       int32
dtype: object
```

# Handling missing values:

## Bookings Data

In [22]:
```
print("Booking Data:")
df_bookings.isnull().sum()
```

```
Booking Data:
```

Out[22]:
```
booking_id                               0
created_at                               0
city                                     0
country                                  0
category_name                            0
product_id                               0
product_name                             0
experience_date                          0
experience_time                          0
customer_id                              0
number_of_guests                         0
customer_country                         0
device                                   0
is_logged_in                             0
web_session_traffic_source           52049
web_session_traffic_medium           52049
web_session_campaign_name           158303
web_session_traffic_origin_country   52202
created_date                             0
exp_date                                 0
created_time                             0
created_hour                             0
exp_time                                 0
exp_hour                                 0
dtype: int64
```

In [23]:
```
round(df_bookings['web_session_traffic_source'].value_counts(normalize=Tr
```

Out[23]:
```
web_session_traffic_source
google                        81.96
(direct)                       9.69
webengage                      4.16
bing                           1.36
dubai_frame                    0.44
                              ...
ebox.co.il                     0.00
ladyandhersweetescapes.com     0.00
thedubaitickets.com            0.00
mail.bg                        0.00
acsds.eubank.kz                0.00
Name: proportion, Length: 458, dtype: float64
```

In [24]:
```
round((df_bookings['web_session_traffic_source'].isnull().sum() / df_book
```
Out[24]:
```
11.37
```

In [25]:
```
df_bookings['web_session_traffic_source'] = df_bookings['web_session_traf
df_bookings['web_session_traffic_medium'] = df_bookings['web_session_traf
```

In [26]:
```
round(df_bookings['web_session_campaign_name'].value_counts(normalize=Tru
```
Out[26]:
```
web_session_campaign_name
Dubai - Burj Khalifa - English - UAE - Search - All - All - cid158
12.66
Dubai - Dubai Frame - English - UAE - Search - All - All - cid1447
8.63
Dubai - Burj Khalifa - Other Languages - UAE - Search - All - All - cid1
58        5.73
Dubai - Things to do - All Languages - UAE - TTD - All - All
4.44
Dubai - Dubai Aquarium - English - UAE - Search - All - All - cid1003 -
Dubai     4.03

...
dubai_newsletter_8
0.00
Dubai - Smash Room - Other Languages - UAE - Search - All - All - cid
0.00
a000CWPncenAE
0.00
r000DW9naenAE
0.00
r007NPGncenAE
0.00
Name: proportion, Length: 1160, dtype: float64
```

In [27]:
```
round((df_bookings['web_session_campaign_name'].isnull().sum() / df_booki
```
Out[27]:
```
34.59
```

In [28]:
```
null_campaigns = df_bookings[df_bookings['web_session_campaign_name'].isn

source_dist = null_campaigns['web_session_traffic_source'].value_counts(n
print("Traffic Sources for Null Campaigns:\n", source_dist)
```
```
Traffic Sources for Null Campaigns:
 web_session_traffic_source
google                          35.843288
unkown/no_web_traffic           32.879352
(direct)                        24.804331
bing                             1.081470
metric.picodi.com                0.861007
                                   ...
mail.aliyun.com                  0.000632
ebox.co.il                       0.000632
ladyandhersweetescapes.com       0.000632
thedubaitickets.com              0.000632
acsds.eubank.kz                  0.000632
Name: proportion, Length: 436, dtype: float64
```

In [29]:
```
df_bookings['web_session_traffic_source'].unique()
```

```
Out[29]: array(['google', 'facebook', 'unkown/no_web_traffic', '(direct)',
                'secure.livechatinc.com', 'webengage', 'headout.kb.help',
                'zendesk', 'bing', 'newsletter', 'duckduckgo', 'rezeem',
                'shareasale', 'dhow-cruise.com', 'facebook.com', 'ampproject.or
         g',
                'groupon.ae', 'ecosia.org', 'tourscanner', 'serverTrigerred',
                'metric.picodi.net', 'dubai_frame', 'miraclegardentickets.com',
                'magdalena_plucińska', 'yahoo', 'uk.search.yahoo.com', 'yandex.r
         u',
                'nm.abv.bg', 'vrparkdubaitickets.com', 'rezeem.com',
                'thedubaiframe.com', 'livechatinc.com', 'l.facebook.com',
                'm.facebook.com', 'aindubai.info', 'dubaiexklusiv',
                'theaindubai.com', 'SilverpopMailing', 'mobile.facebook.com',
                'retailmenot.com', 'promocode.cloud', 'book.imgworldstickets.co
         m',
                'nm20.abv.bg', 'FB', 'rediffmail.com', 'cse.google.com',
                'sociablelabs.com', 'vouchercodesuae.com',
                'topgolfdubaitickets.com', 'tourscanner.com', 'mail.google.com',
                'zerohedge.com', 'instagram.com', 'deref-gmx.net',
                'dontpayfull.com', 'm.nearbyme.io', 'loky.ch', 'zoutons.ae',
                'zimbra.free.fr', 'vero', 'qwant.com', 'search.aol.co.uk',
                'trip101', 'shareasale-analytics.com', 'lm.facebook.com',
                'palmtowerdubai.com', 'poczta.o2.pl', 'offers.com', 'youtube.co
         m',
                'rebajas.guru', 'away.vk.com', 'snapchat.com', 'metric.picodi.co
         m',
                'in.search.yahoo.com', 'the-saudi-hacker.blogspot.com',
                'i-funbox.com', 'auc-excel.officeapps.live.com',
                'search-dra.dt.dbankcloud.com', 'mobilemailer-bs.gmx.net',
                'headout.knoji.com', 'nps', 'evernote.com', 'gabikaremsikova.sk',
                'fr.search.yahoo.com', 'https://www.parisentdecken.de',
                'deref-web.de', 'mail.ru', 'webmail.sfr.fr', 'search.becovi.com',
                '3ds2.checkout.com', '3dverifystc.emcrey.com', 'secure7.arcot.co
         m',
                'ecom.eglobal.com.mx', 'sharjahdesertparktickets.com',
                'authentication.cardinalcommerce.com', 'netsafe.hdfcbank.com',
                'secure5.arcot.com', 'acsoab.com', 'api.checkout.com',
                'acs1.3dsecure.no', 'secureurl.ukr.net',
                'egateway.bankofmaldives.com.mv',
                'mastercardsecurecode.secureacs.com', 'ims.euronet3dsecure.com',
                'inda05.indamail.hu', 'otp.gps.com.bh', 'acs.swisscard.ch',
                'mail.infomaniak.com', 'secure-acs2ui-b1-indmum-mumrdc.wibmo.co
         m',
                'acs1.sbrf.ru', 'secure.3dsib.com', 'acs.bmcebank.ma',
                'acs1.3ds.modirum.com', '3dverify.anb.com.sa',
                'secure.axisbank.com', 'mail.inbox.lv', 'mcconsumer.alahli.com',
                'tinkoff.ru', 'csmu.enstage-sas.com', 'us.search.yahoo.com',
                'abudhabi-tickets.com', 'secure1.axisbank.com', '3ds.cdm.co.ma',
                'br.search.yahoo.com', 'email.seznam.cz', 'securesuite.co.uk',
                'img.ucweb.com', 'citibank.co.in', 'klarna', 'localhost:44117',
                'secure-acs2ui-b1-jak-jakpdc.wibmo.com', 'acs.techcombank.com.v
         n',
                'junglebaytickets.com', 'l.messenger.com', 'wethrift.com',
                'acs2.icicibank.com', 'startpage.com', 'seznam', 'asaan.com',
                'bepguic.npci.org.in', 'arcot.com',
                'cardsecurity.standardchartered.com', 'securesuite.net',
                'coupon.ae', 'couponbricks.com', 'hk.search.yahoo.com',
                'boncode.ae', 'trides-cld.asseco-see.hr', 'l.instagram.com',
                'csch.enstage-sas.com', 'acs-visasecure.acdcproc.com',
                'lg.provenpixel.com', 'vs3dverifybsf.emcrey.com', 'auth.nbo.co.o
```

```
m',
       'book.abudhabi-tickets.com', 'securepayment.meezanbank.com:9612',
       '3ds.icicibank.com', 'zalo', 'family.ctbcbank.com',
       'email.t-online.de', 'safekey-1.americanexpress.com',
       'dealspotr.com', 'x.cna-tech.com', 'YouTube.com',
       'bmail.uol.com.br', 'images.search.yahoo.com', 'trustpilot',
       'email.telstra.com', 'afmail.uol.com.br',
       'statics.teams.cdn.office.net', 'outlook.live.com',
       'couponcodesme.com', 'vbv2.eahli.com', 'ch.search.yahoo.com',
       'grabon.in', 'visa-cipher2.gw.zetapay.in',
       'https://linktr.ee/dubaionboard', 't-mail.centrum.sk',
       '3dsbiacs.bank.sbi', 'links.rediff.com', 'thedubaizipline.com',
       'joinhoney.com', 'googleads.g.doubleclick.net',
       'acs.hanacard.co.kr', 'mycardsecure.com',
       'indusindbank-mas102-cipher2-mum.gw.zetapay.in', 'pay.google.co
m',
       'perksatwork.com', 'poczta.onet.pl', 't-mail.centrum.cz',
       'otlobcoupon.com', 'click.mail.ru', 'app.deel.com', 'digi_mark',
       'deref-1und1-02.de', 'email.inbox.lv', 'mail.uol.com.br',
       '(not set)', 'web-mail.laposte.net', 'wanderlog.com',
       'headout.looker.com', 'redirect.viglink.com', 'yandex.by',
       '5fb50a613373ed3217518394d5a059c0.safeframe.googlesyndication.co
m',
       'https://ticketcombo.net/', 'suche.web.de', 'mail01.orange.fr',
       'acs.icicibank.com', 'authenticationweb.cartoes-itau.com.br',
       'acs2.sbrf.ru', 'acs.privatbank.ua', 'secure.tinkoff.ru',
       '3dsecure1.icicibank.com', '3dsp.vtb.ru',
       'ipcacs.sber-bank.by:9753', '3dsecure.garanti.com.tr',
       '3dsecure.icicibank.com', 'acs2.3ds.modirum.com', '3ds.payment.r
u',
       'mobilemailer-bap.gmx.net',
       'indusindbank-visa102-cipher2-mum.gw.zetapay.in',
       'nationstrust.com', 'cardsecure.kkb.kz', 'webmail1.sunrise.ch',
       'themeparkstickets.com', 'securepayments.unionbankofindia.co.in',
       'acs.3dsecure.az', 'acs-idcheck.acdcproc.com',
       'secure.iraqegate.iq', 'acs.kapital24.uz:9602', 'couponchief.co
m',
       'vbv.scb.co.th', 'mla', 'skiptheline', 'evgeny-nadymov.github.i
o',
       'exmail.qq.com', 'cloudsdeal.com', 'https://skiptheline.ticket
s/',
       'search-dre.dt.dbankcloud.com', 'be.search.yahoo.com',
       'accounts.google.com', 'pca3ds.gbp.ma:4443',
       'login.microsoftonline.com', '3dverifyalinma.emcrey.com',
       'secure2.arcot.com', '3dauth.mbu.hr', 'tw.search.yahoo.com',
       'linktr.ee', 'poczta.interia.pl', '10.0.0.8', 'go-go.tech',
       'cloudmail.concept-its.co.uk', 'vmail.centrum.cz',
       'epayiss.thecitybank.com', 'yandex', 'url.qmail.com',
       'nm80.abv.bg', 'burjkhalifaticketsuk', 'info.com',
       'pl.search.yahoo.com', 'skiptheline.tickets', 'secure4.arcot.co
m',
       'secure-acs2ui-b1-indblr-blrtdc.wibmo.com', 'acs.mashreq.com',
       'alignet-acs.com', 'acs2.swedbank.se', 'adfs.voestalpine.com',
       'mail.centrum.cz', 'secure6.arcot.com', 'FuckOff',
       'deref-mail.com', 'headout.zendesk.com', 'otp.uzcard.uz',
       'secure22gw.ro', 'https://www.couponplusdeal.com/',
       'geschuetztkaufen2.commerzbank.de', 'coda.io',
       'sdc-yb.enstage-sas.com', 'holidify.com', 'youtube',
       'gladebrookcapital.com', 'qq.com', 'wx.mail.qq.com', 'start.co.i
l',
       '200journeys', 'www58.bb.com.br', 'linkin.bio',
       'dbsbank.euronet3dsecure.com', '192.168.90.35:15871',
       'm.exmail.qq.com', 'acs.bankofindia.co.in', 'link.avito.ru',
       'acs.s2mgcc.com', 'authentication2.six-group.com',
       'mcsthreed.baj.com.sa', 'webmail.tim.it', 'acssv.ckb.me',
       'rich-v01.bluewin.ch', 'acs.burgan.com', '3ds.qnb.com',
       'auth2.securtxn.com', 't.co', 'pdc-yb.enstage-sas.com',
       'ca.search.yahoo.com', 'trustpilot.com', 'no.search.yahoo.com',
       'trc.taboola.com', 'skywalkdubaitickets.com',
       'palmtowertickets.com', 'thearchtickets.com', 'skywalkdubai.com',
       'hattawadihubtickets.com', 'mobilemailer-bap.web.de',
       'couponskiss.com', 'tsys.arcot.com',
       'mastercardidentitycheck.sparkassen-kreditkarten.de',
       'euc-excel.officeapps.live.com', 'poczta.wp.pl',
       'postbank-3ds-bxl.wlp-acs.com', 'pay.activa-card.com',
       'e-secure.bop.ps', 'vbv.samsungcard.co.kr',
       'acs2.bpcprocessing.com', 'mail.aliyun.com', 'ebox.co.il',
       'ladyandhersweetescapes.com', 'gatekeeperapp.net',
       'thedubaitickets.com', 'mail.bg', 'jac.yahoosandbox.com',
       'acm2.eim.ae', 'poczta.put.poznan.pl', 'gardenglowtickets.com',
       '3ds.banquemisr.com:4443',
       'verifiedbyvisa.acs.touchtechpayments.com', 'acs1.itcbd.com:1828
6',
       'it.search.yahoo.com', 'secureauthentication2.citibank.com',
       'karmanow.com', 'wildwaditickets.com', 'dolphinariumtickets.com',
       'suche.t-online.de', 'baidu', 'm.baidu.com',
       'mybrowser-search.com', 'allcoupons.ae', 'trust.s2mgcc.com',
       'cbbankcard.net', 'acs4.sbrf.ru', 'r.couponasion.com',
       'webmail2.sunrise.ch', 'mail.qq.com',
       'zombieapocalypseparktickets.com', 'r.srvtrck.com',
       'webmail1n.orange.fr', 'burjkhalifatickets.co.uk',
       'keep.google.com', 'acs1.icicibank.com',
       'idcheck.acs.touchtechpayments.com', 'acs.bkm.com.tr',
       'mail.azet.sk', 'acs.upc.ua', '3dsecure.raiffeisenbank.rs',
       'icicibank.com', 'api.acs.opentech.com', 'sas.redsys.es',
       'b4-pdc.enstage-sas.com', 'verifiedbyvisa.secureacs.com',
       'acs.ipakyulibank.uz:7443', 'bepguih.npci.org.in', 'acs.hnb.lk',
       'acs.mepspay.com:445', '3ds.unibank.az', 'mail.tiscali.it',
       'acs12.bmcebank.ma', 'secureshopping.usaa360.com', 'm.abv.bg',
       'acs.kbcard.com', '3dsecure.kapitalbank.az', 'acs3.3dsecure.no',
       'acs.nedsecure.co.za', 'web.telegram.org', 'weixin110.qq.com',
       'webmail.worsfoldgregg.com',
       'bf344e2deb6f77a23329c936cf1d6cc7.safeframe.googlesyndication.co
m',
       'lightmailer-bap.web.de',
       '420fae45355f44907165e7be0c5dc255.safeframe.googlesyndication.co
m',
       'm.youtube.com', 'petalsearch.com', '10minutemail.com',
       'plumbucket.com', 'se.search.yahoo.com', 'acs2.arca.am',
       'acsweb-pa.dnp-cdms.jp', 'german-3ds-bxl.wlp-acs.com',
       'dubai-experience.com', 'email17.godaddy.com',
       'sg.search.yahoo.com', '3dsec.cardcenter.ch',
       'ecommerce.aps.iq:4443', 'google.ae', 'acs-v2.fh.ae',
       'amail.centrum.sk', 't.post.sme.sk', 'ph.search.yahoo.com',
       '3dverifyalbilad.emcrey.com', 'acs.unifiedpaymentsnigeria.com',
       'yandex.kz', 'b8-pdc.enstage-sas.com', 'search.aol.com',
       'everysaving.sg', 'epp.khanbank.com', 'ecom.pbebank.com',
       'secure.3ds.cornercard.ch', 'https://urlaubindubai.com',
       'l.workplace.com', 'inda02.indamail.hu', 'webengage/',
       'engine.presearch.org', 'search.brave.com', '3dsecure.bcc.kz:344
```

```
3',
       'www2.acs.bdo.com.ph', 'm.gsearch.co', 'mx.search.yahoo.com',
       'rdtds.net',
       '3a71a79182989caeec1676758cd33230.safeframe.googlesyndication.co
m',
       '3ds.vtb.ru', '3dsecure.slsp.sk', 'acs.cihanbank.com',
       'poshukach.com', '3d-secure2.sbanken.no', 'acs2.nbu.uz',
       'at.search.yahoo.com', '3ds.altynbank.kz:3443',
       '3dsecure.raiffeisen.al', 'authentication1.six-group.com',
       'acs.quipugmbh.com', 'picodi.com', 'mymail.myt.mu',
       'blondontheroad.com', 'blog.wego.com',
       '6d7dc6237d1e4a408f4846b01257e2bb.safeframe.googlesyndication.co
m',
       'googleadservices.com', 'docs.google.com',
       'secureauthentication.citibank.com', 'mail.walla.co.il',
       'https://www.viaggiare_low_cost.it/', 'linkedin.com',
       'acs1.sparebank1.no', 'de.search.yahoo.com', 'poczta.gazeta.pl',
       'link.edgepilot.com', 'webmail.freenet.de', 'blog_iframe',
       'oceanhero.today', 'travelmasterpieces.com',
       'authentication-acs.marqeta.com', 'vsconsumer.alahli.com',
       'acsds.eubank.kz'], dtype=object)
```

In [30]:
```python
def refine_campaign(row):
    source = str(row['web_session_traffic_source']).lower()

    if any(s in source for s in ['google', 'bing', 'yahoo', 'duckduckgo']
        return 'organic_search_no_campaign'

    if '(direct)' in source:
        return 'direct_no_campaign'

    return 'unattributed'

mask = df_bookings['web_session_campaign_name'].isnull()
df_bookings.loc[mask, 'web_session_campaign_name'] = df_bookings[mask].ap

round(df_bookings['web_session_campaign_name'].value_counts(normalize=Tru
```

Out[30]:
```
web_session_campaign_name
unattributed
13.02
organic_search_no_campaign
12.99
direct_no_campaign
8.58
Dubai - Burj Khalifa - English - UAE - Search - All - All - cid158
8.28
Dubai - Dubai Frame - English - UAE - Search - All - All - cid1447
5.64

...
Dubai - Al Maha Desert Resort & Spa - English - UAE - Search - All - All
- cid1074 - Dubai     0.00
r000WUIncenUS
0.00
cashback_vatican_en
0.00
r0008FMnaenAU
0.00
a006ANJraenCA
0.00
Name: proportion, Length: 1163, dtype: float64
```

## web_session_traffic_origin_country:

In [32]:
```
round(df_bookings['web_session_traffic_origin_country'].value_counts(norm
```

Out[32]:
```
web_session_traffic_origin_country
United Arab Emirates    80.79
India                    3.23
United Kingdom           1.88
United States            1.88
Saudi Arabia             1.23
                        ...
Western Sahara           0.00
American Samoa           0.00
Bahamas                  0.00
Lesotho                  0.00
Liechtenstein            0.00
Name: proportion, Length: 180, dtype: float64
```

In [33]:
```
df_bookings['web_session_traffic_origin_country'] = df_bookings['web_sess
```

In [34]:
```
df_bookings.isna().sum()
```

Out[34]:
```
booking_id                          0
created_at                          0
city                                0
country                             0
category_name                       0
product_id                          0
product_name                        0
experience_date                     0
experience_time                     0
customer_id                         0
number_of_guests                    0
customer_country                    0
device                              0
is_logged_in                        0
web_session_traffic_source          0
web_session_traffic_medium          0
web_session_campaign_name           0
web_session_traffic_origin_country  0
created_date                        0
exp_date                            0
created_time                        0
created_hour                        0
exp_time                            0
exp_hour                            0
dtype: int64
```

## Events Data:

In [36]:
```
df_events.isna().sum()
```

Out[36]:
```
event_id                                 0
event_timestamp                          0
event_name                               0
customer_id                              0
session_id                               0
product_id                               0
booking_id                          196014
event_type                               0
position                                 0
page_type                                0
device                                   0
is_logged_in                             0
smart_rec_enabled                        0
rec_impressions                          0
rec_clicks                               0
rec_slot_position                   238623
search_filters_used                      0
num_filters_applied                      0
search_query_length                      0
web_session_traffic_origin_country   31055
web_session_traffic_source           30947
web_session_traffic_medium           30947
web_session_campaign_name           103179
city                                     0
is_first_time_customer                   0
customer_lifetime_bookings               0
experiment_bucket                        0
active_discount_flag                     0
supply_tier                              0
event_date                               0
event_time                               0
event_hour                               0
dtype: int64
```

In [37]:
```
df_events['booking_id'] = df_events['booking_id'].fillna('no_booking_done
```

In [38]:
```
round((df_events['rec_slot_position'].isna().sum() / df_events.shape[0])*
```

Out[38]:
```
89.13
```

In [39]:
```
round(df_events[df_events['rec_slot_position'].isna()]['smart_rec_enabled
```

Out[39]:
```
smart_rec_enabled
1    50.6
0    49.4
Name: proportion, dtype: float64
```

In [40]:
```
round(df_events[df_events['rec_slot_position'].isna()]['rec_clicks'].valu
```

Out[40]:
```
rec_clicks
0    100.0
Name: proportion, dtype: float64
```

This means rec_slot_position is only tracked for events when the recommendation is clicked. filling null values with 0 to represent no click happened

In [42]:
```
df_events['rec_slot_position'] = df_events['rec_slot_position'].fillna(0)
```

In [43]:
```
df_events['web_session_traffic_source'] = df_events['web_session_traffic_
df_events['web_session_traffic_medium'] = df_events['web_session_traffic_
```

In [44]:
```
df_events.isnull().sum()
```

Out[44]:
```
event_id                                 0
event_timestamp                          0
event_name                               0
customer_id                              0
session_id                               0
product_id                               0
booking_id                               0
event_type                               0
position                                 0
page_type                                0
device                                   0
is_logged_in                             0
smart_rec_enabled                        0
rec_impressions                          0
rec_clicks                               0
rec_slot_position                        0
search_filters_used                      0
num_filters_applied                      0
search_query_length                      0
web_session_traffic_origin_country   31055
web_session_traffic_source               0
web_session_traffic_medium               0
web_session_campaign_name           103179
city                                     0
is_first_time_customer                   0
customer_lifetime_bookings               0
experiment_bucket                        0
active_discount_flag                     0
supply_tier                              0
event_date                               0
event_time                               0
event_hour                               0
dtype: int64
```

In [45]:
```
def refine_campaign(row):
    source = str(row['web_session_traffic_source']).lower()

    if any(s in source for s in ['google', 'bing', 'yahoo', 'duckduckgo']
        return 'organic_search_no_campaign'

    if '(direct)' in source:
        return 'direct_no_campaign'

    return 'unattributed'

mask = df_events['web_session_campaign_name'].isnull()
df_events.loc[mask, 'web_session_campaign_name'] = df_events[mask].apply(

round(df_events['web_session_campaign_name'].value_counts(normalize=True)
```

Out[45]:
```
web_session_campaign_name
organic_search_no_campaign                                                  1
4.69
unattributed                                                                1
2.85
direct_no_campaign                                                          1
1.00
Dubai - Burj Khalifa - English - UAE - Search - All - All - cid158
8.79
Dubai - Dubai Frame - English - UAE - Search - All - All - cid1447
6.38

...
r001Z3ArcenIN
0.00
a006ALKncenAE
0.00
r001T9RnaenUS
0.00
a001VVLncdeDE
0.00
r0001C9ncenAE
0.00
Name: proportion, Length: 576, dtype: float64
```

In [46]:
```
df_events['web_session_traffic_origin_country'] = df_events['web_session_
```

In [47]:
```
df_events.isnull().sum()
```

```
Out[47]: event_id                                0
         event_timestamp                         0
         event_name                              0
         customer_id                             0
         session_id                              0
         product_id                              0
         booking_id                              0
         event_type                              0
         position                                0
         page_type                               0
         device                                  0
         is_logged_in                            0
         smart_rec_enabled                       0
         rec_impressions                         0
         rec_clicks                              0
         rec_slot_position                       0
         search_filters_used                     0
         num_filters_applied                     0
         search_query_length                     0
         web_session_traffic_origin_country      0
         web_session_traffic_source              0
         web_session_traffic_medium              0
         web_session_campaign_name               0
         city                                    0
         is_first_time_customer                  0
         customer_lifetime_bookings              0
         experiment_bucket                       0
         active_discount_flag                    0
         supply_tier                             0
         event_date                              0
         event_time                              0
         event_hour                              0
         dtype: int64
```

```
In [48]: df_bookings.isna().sum()
```

Out[48]:
```
booking_id                        0
created_at                        0
city                              0
country                           0
category_name                     0
product_id                        0
product_name                      0
experience_date                   0
experience_time                   0
customer_id                       0
number_of_guests                  0
customer_country                  0
device                            0
is_logged_in                      0
web_session_traffic_source        0
web_session_traffic_medium        0
web_session_campaign_name         0
web_session_traffic_origin_country    0
created_date                      0
exp_date                          0
created_time                      0
created_hour                      0
exp_time                          0
exp_hour                          0
dtype: int64
```

In [49]:
```
drop_cols = ['experience_date', 'experience_time', 'created_at']
df_bookings = df_bookings.drop(columns=drop_cols)
df_bookings
```

Out[49]:

| | booking_id | city | country | category_name | produ |
|---|---|---|---|---|---|
| **0** | bid_+1fN3xKE6A7bLb3Mb8EHug== | Dubai | United Arab Emirates | Dubai Dhow Cruises | |
| **1** | bid_oeNEooQN+xls8LMa8z7sMg== | Dubai | United Arab Emirates | Dubai Dhow Cruises | |
| **2** | bid_+1E+CdL+uNH/9LMgCTphtw== | Dubai | United Arab Emirates | Ferrari World Tickets | |
| **3** | bid_o55xFpl5f2ndjyBjWqgq/g== | Dubai | United Arab Emirates | Museum of Illusions Tickets | |
| **4** | bid_Qj1OEgYvKdCD1YXJzq2+PQ== | Dubai | United Arab Emirates | Museum of Illusions Tickets | |
| **...** | ... | ... | ... | ... | |
| **457615** | bid_8DnULqWYueMBbBeQwPtoWw== | Dubai | United Arab Emirates | Wild Wadi Tickets | |
| **457616** | bid_bHv5RVfuypSgggjvoMUZtQ== | Dubai | United Arab Emirates | Wild Wadi Tickets | |
| **457617** | bid_IjcnoEBVJVpVOw8Lb3zo8w== | Dubai | United Arab Emirates | Wild Wadi Tickets | |
| **457618** | bid_u1BRd6nDyiHnOXBrsCbIHQ== | Dubai | United Arab Emirates | Wild Wadi Tickets | |
| **457619** | bid_0B95GBlvdu8InpQTS9sOHg== | Dubai | United Arab Emirates | Wild Wadi Tickets | |

457620 rows × 21 columns

# User Personas:

## Lead Days:

In [52]:

```
df_bookings['created_date'] = pd.to_datetime(df_bookings['created_date'],
df_bookings['exp_date'] = pd.to_datetime(df_bookings['exp_date'], errors=
df_bookings['lead_days'] = (df_bookings['exp_date'] - df_bookings['create
df_bookings
```

Out[52]:

| | booking_id | city | country | category_name | produ |
|---|---|---|---|---|---|
| **0** | bid_+1fN3xKE6A7bLb3Mb8EHug== | Dubai | United Arab Emirates | Dubai Dhow Cruises | |
| **1** | bid_oeNEooQN+xls8LMa8z7sMg== | Dubai | United Arab Emirates | Dubai Dhow Cruises | |
| **2** | bid_+1E+CdL+uNH/9LMgCTphtw== | Dubai | United Arab Emirates | Ferrari World Tickets | |
| **3** | bid_o55xFpl5f2ndjyBjWqgq/g== | Dubai | United Arab Emirates | Museum of Illusions Tickets | |
| **4** | bid_Qj1OEgYvKdCD1YXJzq2+PQ== | Dubai | United Arab Emirates | Museum of Illusions Tickets | |
| **...** | ... | ... | ... | ... | |
| **457615** | bid_8DnULqWYueMBbBeQwPtoWw== | Dubai | United Arab Emirates | Wild Wadi Tickets | |
| **457616** | bid_bHv5RVfuypSgggjvoMUZtQ== | Dubai | United Arab Emirates | Wild Wadi Tickets | |
| **457617** | bid_IjcnoEBVJVpVOw8Lb3zo8w== | Dubai | United Arab Emirates | Wild Wadi Tickets | |
| **457618** | bid_u1BRd6nDyiHnOXBrsCbIHQ== | Dubai | United Arab Emirates | Wild Wadi Tickets | |
| **457619** | bid_0B95GBlvdu8InpQTS9sOHg== | Dubai | United Arab Emirates | Wild Wadi Tickets | |

457620 rows × 22 columns

In [53]:

```
lead_time_dist = (pd.DataFrame(df_bookings['lead_days'].value_counts(norm
lead_time_dist[1:20]
```

Out[53]:

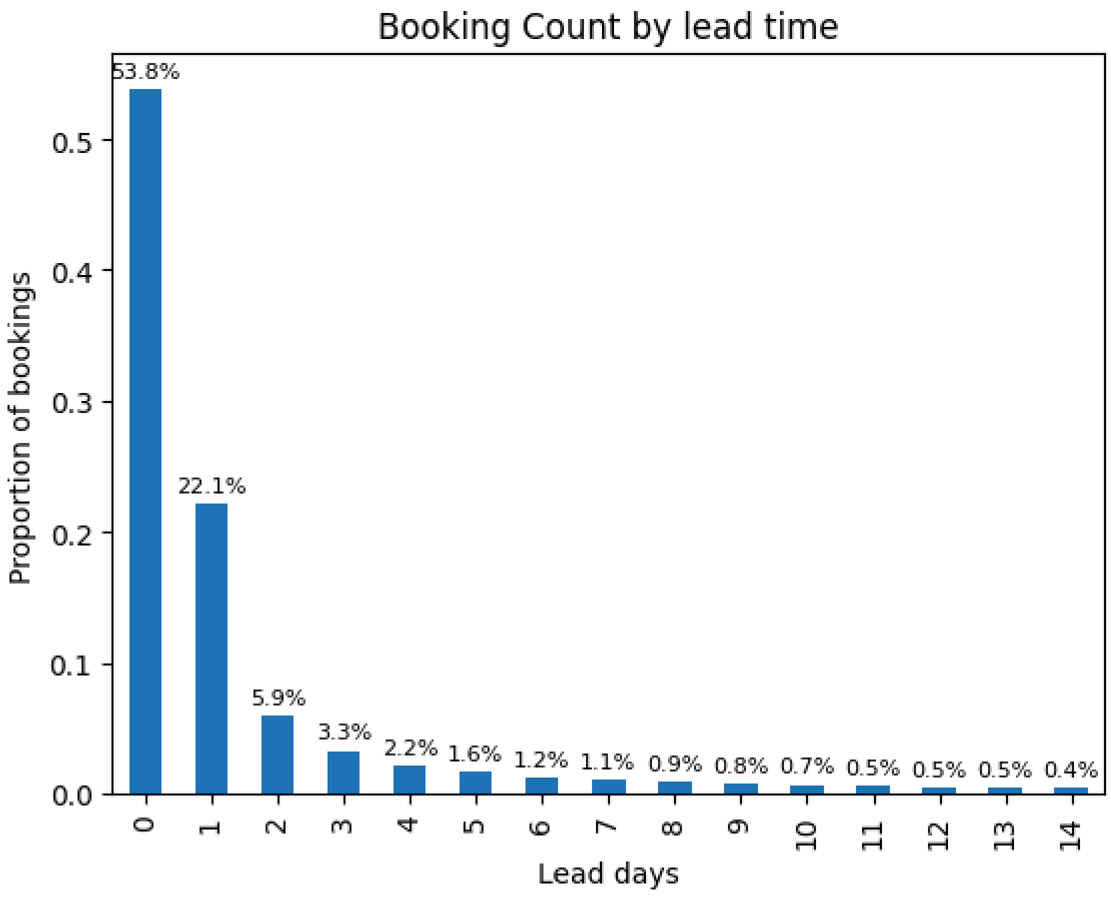
| lead_days | proportion |
|---|---|
| 0 | 53.8 |
| 1 | 75.9 |
| 2 | 81.8 |
| 3 | 85.1 |
| 4 | 87.3 |
| 5 | 88.9 |
| 6 | 90.1 |
| 7 | 91.2 |
| 8 | 92.1 |
| 9 | 92.9 |
| 10 | 93.6 |
| 11 | 94.1 |
| 12 | 94.6 |
| 13 | 95.1 |
| 14 | 95.5 |
| 15 | 95.9 |
| 16 | 96.2 |
| 17 | 96.5 |
| 18 | 96.7 |

In [54]:
```
round(df_bookings['lead_days'].value_counts(normalize=True)*100,1).head(1
```

Out[54]:
```
lead_days
0    53.8
1    22.1
2     5.9
3     3.3
4     2.2
5     1.6
6     1.2
7     1.1
8     0.9
9     0.8
Name: proportion, dtype: float64
```

In [55]:
```
lead_share = (
    df_bookings['lead_days']
    .value_counts(normalize=True)
    .sort_index()
)[1:]

ax = lead_share.loc[:14].plot(kind='bar')
```

```
plt.ylabel('Proportion of bookings')
plt.xlabel('Lead days')
plt.title('Booking Count by lead time')

for p in ax.patches:
    value = p.get_height()
    ax.annotate(
        f'{value*100:.1f}%',
        (p.get_x() + p.get_width() / 2, value),
        ha='center',
        va='bottom',
        fontsize=8,
        xytext=(0, 3),
        textcoords='offset points'
    )

plt.show()
```



0 : same day

1 : next day

2-4 : short planner

5-7 : moderate planner

7+ : advance planner

In [57]:
```
def lead_day_bucket(x):
    if x == 0:
        return 'same_day'
    elif x == 1:
        return 'next_day'
    elif x <=4:
        return 'short_planner'
    elif x <= 7:
        return 'moderate_planner'
    else:
        return 'advance_planner'

df_bookings['lead_days_bin'] = df_bookings['lead_days'].apply(lead_day_bu

round(df_bookings['lead_days_bin'].value_counts(normalize=True)*100,1)
```

Out[57]:

```
lead_days_bin
same_day            53.8
next_day            22.1
short_planner       11.3
advance_planner      8.8
moderate_planner     3.9
Name: proportion, dtype: float64
```

## Guest Size:

In [59]:

```
round(df_bookings['number_of_guests'].value_counts(normalize=True)*100,2)
```

Out[59]:

```
number_of_guests
1     12.99
2     39.10
3     18.27
4     15.50
5      6.64
6      3.41
7      1.73
8      1.06
9      0.51
10     0.49
11     0.09
12     0.06
13     0.04
14     0.03
15     0.02
16     0.01
17     0.01
18     0.00
19     0.00
20     0.02
Name: proportion, dtype: float64
```

In [60]:

```
round(df_bookings['number_of_guests'].value_counts(normalize=True)*100,2)
```

Out[60]:
```
number_of_guests
1      12.99
2      52.09
3      70.36
4      85.86
5      92.50
6      95.91
7      97.64
8      98.70
9      99.21
10     99.70
11     99.79
12     99.85
13     99.89
14     99.92
15     99.94
16     99.95
17     99.96
18     99.96
19     99.96
20     99.98
Name: proportion, dtype: float64
```

### 1 : solo

### 2 : couple

### 3-4 : small group

### 5+ : large group

In [62]:
```python
def guest_bin(x):
    if x == 1:
        return 'solo'
    elif x == 2:
        return 'couple'
    elif x <= 4:
        return 'small_group'
    else:
        return 'large_group'

df_bookings['guest_size_bin'] = df_bookings['number_of_guests'].apply(gue
round(df_bookings['guest_size_bin'].value_counts(normalize=True)*100,1)
```

Out[62]:
```
guest_size_bin
couple         39.1
small_group    33.8
large_group    14.1
solo           13.0
Name: proportion, dtype: float64
```

## Booking Count:

In [64]:
```python
df_bookings[['customer_id', 'booking_id']].value_counts()
```

Out[64]:
```
customer_id                   booking_id
cus_+++2FLNoNNnNp3D4uV368Q==  bid_0Vnytef+E3Yiog8EGXuL9A==    1
cus_eXjYrt+mvAq6s0iPqaFRzQ==  bid_TGW4XN8k+Pim8fyIAPa/fA==    1
cus_eXnZgtwMIUPfhAadTuEeVg==  bid_xF01wO1IM4gDQ8GqRfSN4Q==    1
cus_eXmrRrwpE21Ax90IP5HgTg==  bid_0ZpFjJhJccOLYDvPq3lvmA==    1
cus_eXmLapxrh4kz8ScPqZ3wOw==  bid_f6o6D6//4NHe8LMppoveww==    1
                                                             ..
cus_J1xh5re5U61k1PxWiDdkdQ==  bid_JvH+ljmfqSKfTEuj0HvIdQ==    1
cus_J1xIxbhLuEzAktv0dqe9sA==  bid_UV7/Q5NDoOlEkcOC8YO74g==    1
cus_J1wDgfFgq/58Zc4+ZquNCA==  bid_hJ6R3IEaesr5aNzEH5eFQA==    1
cus_J1vmxOlVxpz93vL6dSuvQw==  bid_5XdC4wXT807oJBt3+B7X4w==    1
cus_zzygHAYXL/X1fa+ZRjD++Q==  bid_qxm4iQhHAD25z2Tt0mNocQ==    1
Name: count, Length: 457620, dtype: int64
```

In [65]:
```python
df_bookings['customer_booking_count'] = (
    df_bookings
    .groupby('customer_id')['booking_id']
    .transform('nunique')
)
customer_booking_count = df_bookings[['customer_id', 'customer_booking_co
customer_booking_count.drop_duplicates(inplace=True)
round(customer_booking_count['customer_booking_count'].value_counts(norma
```

Out[65]:
```
customer_booking_count
1     73.9
2     17.7
3      4.9
4      1.9
5      0.8
6      0.4
7      0.2
8      0.1
9      0.1
10     0.0
11     0.0
12     0.0
13     0.0
14     0.0
15     0.0
16     0.0
17     0.0
19     0.0
22     0.0
21     0.0
Name: proportion, dtype: float64
```

In [66]:
```python
def booking_count_bin(x):
    if x == 1:
        return 'single_booking'
    elif x == 2:
        return 'repeat_once'
    else:
        return 'multiple repeats'

df_bookings['booking_count_bin'] = df_bookings['customer_booking_count'].
df_bookings.groupby('booking_count_bin')['customer_id'].nunique()
```

Out[66]:
```
booking_count_bin
multiple repeats     26951
repeat_once          56421
single_booking      235595
Name: customer_id, dtype: int64
```

Point to be noted: Booking count is not a direct user persona as its not answering any how or who about the user. its a medium that we can use as an observed signal within personas.

## Validations:

In [69]:
```
df_bookings['core_persona'] = (df_bookings['lead_days_bin'] + ' | ' + df_
round(df_bookings['core_persona'].value_counts(normalize=True)*100,1)
```

Out[69]:
```
core_persona
same_day | couple                  20.9
same_day | small_group             18.3
next_day | couple                   8.7
same_day | large_group              7.4
next_day | small_group              7.3
same_day | solo                     7.2
short_planner | couple              4.4
short_planner | small_group         3.8
advance_planner | couple            3.5
next_day | large_group              3.1
advance_planner | small_group       2.9
next_day | solo                     2.9
short_planner | large_group         1.7
moderate_planner | couple           1.5
short_planner | solo                1.4
moderate_planner | small_group      1.4
advance_planner | large_group       1.3
advance_planner | solo              1.0
moderate_planner | large_group      0.6
moderate_planner | solo             0.5
Name: proportion, dtype: float64
```

## Correlation between group size and planning span:

In [71]:
```
pd.crosstab(
    df_bookings['lead_days_bin'],
    df_bookings['guest_size_bin'],
    normalize='all'
).round(3)*100
```

Out[71]:

| guest_size_bin | couple | large_group | small_group | solo |
|---|---|---|---|---|
| **lead_days_bin** | | | | |
| **advance_planner** | 3.5 | 1.3 | 2.9 | 1.0 |
| **moderate_planner** | 1.5 | 0.6 | 1.4 | 0.5 |
| **next_day** | 8.7 | 3.1 | 7.3 | 2.9 |
| **same_day** | 20.9 | 7.4 | 18.3 | 7.2 |
| **short_planner** | 4.4 | 1.7 | 3.8 | 1.4 |

## Most of the bookings happen within the same day for all group size (~54%).

## followed by those who plan just one day ago (~20%)

## showing most of the customers are urgent planners irrespective of group size

## Median gap days between personas:

In [74]:
```
spacing = (
    df_bookings
    .groupby(['core_persona', 'customer_id'])['created_date']
    .apply(lambda x: x.sort_values().diff().dt.days.median())
    .reset_index(name='median_gap_days')
)

spacing = spacing.merge(
    df_bookings[['customer_id','booking_count_bin']].drop_duplicates(),
    on='customer_id',
    how='left'
)

spacing.groupby(
    ['core_persona','booking_count_bin']
)['median_gap_days'].median().unstack()
```

Out[74]:

| booking_count_bin<br>core_persona | multiple repeats | repeat_once | single_booking |
|---|---|---|---|
| advance_planner \| couple | 0.0 | 0.0 | NaN |
| advance_planner \| large_group | 0.0 | 0.0 | NaN |
| advance_planner \| small_group | 0.0 | 0.0 | NaN |
| advance_planner \| solo | 0.0 | 0.0 | NaN |
| moderate_planner \| couple | 0.0 | 0.0 | NaN |
| moderate_planner \| large_group | 0.0 | 0.0 | NaN |
| moderate_planner \| small_group | 0.0 | 0.0 | NaN |
| moderate_planner \| solo | 0.0 | 0.0 | NaN |
| next_day \| couple | 0.5 | 0.0 | NaN |
| next_day \| large_group | 0.5 | 0.0 | NaN |
| next_day \| small_group | 1.0 | 0.0 | NaN |
| next_day \| solo | 0.0 | 0.0 | NaN |
| same_day \| couple | 1.5 | 0.0 | NaN |
| same_day \| large_group | 2.0 | 0.0 | NaN |
| same_day \| small_group | 2.0 | 1.0 | NaN |
| same_day \| solo | 0.0 | 0.0 | NaN |
| short_planner \| couple | 0.0 | 0.0 | NaN |
| short_planner \| large_group | 0.0 | 0.0 | NaN |
| short_planner \| small_group | 0.0 | 0.0 | NaN |
| short_planner \| solo | 0.0 | 0.0 | NaN |

### Most of the personas who have 2 bookings, complete their bookings within the same day excpet for same_day | small_group.

### For those having 3 or more bookings, among them only urgent planners seems to have their bookings distributed across multiple days, where next day travellers have a median gap of 0.5 days and same day has 1.25

## Product Category Exploration by Persona:

In [77]:
```
persona_cust = (
    df_bookings
    .groupby(['core_persona','customer_id'])
    .agg(
        category_count=('category_name','nunique'),
        booking_count=('booking_id','nunique')
    )
    .reset_index()
)

persona_cust.groupby('core_persona')['category_count'].describe().sort_va
```

Out[77]:

| core_persona | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **advance_planner \| large_group** | 4134.0 | 1.291485 | 0.794635 | 1.0 | 1.0 | 1.0 | 1.0 | 11.0 |
| **advance_planner \| small_group** | 9192.0 | 1.290796 | 0.810905 | 1.0 | 1.0 | 1.0 | 1.0 | 12.0 |
| **advance_planner \| couple** | 11110.0 | 1.278848 | 0.811195 | 1.0 | 1.0 | 1.0 | 1.0 | 12.0 |
| **advance_planner \| solo** | 3303.0 | 1.157433 | 0.577211 | 1.0 | 1.0 | 1.0 | 1.0 | 8.0 |
| **moderate_planner \| large_group** | 2213.0 | 1.126073 | 0.457915 | 1.0 | 1.0 | 1.0 | 1.0 | 6.0 |
| **moderate_planner \| small_group** | 4969.0 | 1.121755 | 0.446253 | 1.0 | 1.0 | 1.0 | 1.0 | 8.0 |
| **moderate_planner \| couple** | 5510.0 | 1.116334 | 0.420571 | 1.0 | 1.0 | 1.0 | 1.0 | 6.0 |
| **same_day \| small_group** | 71450.0 | 1.114080 | 0.409326 | 1.0 | 1.0 | 1.0 | 1.0 | 10.0 |
| **same_day \| couple** | 80919.0 | 1.110122 | 0.400013 | 1.0 | 1.0 | 1.0 | 1.0 | 10.0 |
| **same_day \| large_group** | 28961.0 | 1.103104 | 0.388550 | 1.0 | 1.0 | 1.0 | 1.0 | 11.0 |
| **short_planner \| small_group** | 14505.0 | 1.092382 | 0.373026 | 1.0 | 1.0 | 1.0 | 1.0 | 10.0 |
| **short_planner \| large_group** | 6512.0 | 1.090448 | 0.367996 | 1.0 | 1.0 | 1.0 | 1.0 | 6.0 |
| **short_planner \| couple** | 16857.0 | 1.088094 | 0.359806 | 1.0 | 1.0 | 1.0 | 1.0 | 8.0 |
| **moderate_planner \| solo** | 1711.0 | 1.078901 | 0.335387 | 1.0 | 1.0 | 1.0 | 1.0 | 5.0 |
| **next_day \| couple** | 34325.0 | 1.076300 | 0.327665 | 1.0 | 1.0 | 1.0 | 1.0 | 8.0 |
| **next_day \| small_group** | 29352.0 | 1.075293 | 0.327059 | 1.0 | 1.0 | 1.0 | 1.0 | 9.0 |
| **next_day \| large_group** | 12431.0 | 1.070228 | 0.328787 | 1.0 | 1.0 | 1.0 | 1.0 | 8.0 |
| **same_day \| solo** | 27588.0 | 1.066152 | 0.308427 | 1.0 | 1.0 | 1.0 | 1.0 | 7.0 |
| **short_planner \| solo** | 5215.0 | 1.057910 | 0.311064 | 1.0 | 1.0 | 1.0 | 1.0 | 12.0 |
| **next_day \| solo** | 11399.0 | 1.046144 | 0.256812 | 1.0 | 1.0 | 1.0 | 1.0 | 8.0 |

### Insights: Bookings are heavily single-experience driven, regardless of planning span or group size

## Booking Count intensity by persona:

```
In [80]: persona_cust.groupby('core_persona')['booking_count'].describe().sort_val
```

Out[80]:

| core_persona | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **advance_planner \| large_group** | 4134.0 | 1.480406 | 1.090610 | 1.0 | 1.0 | 1.0 | 2.0 | 22.0 |
| **advance_planner \| small_group** | 9192.0 | 1.465078 | 1.000967 | 1.0 | 1.0 | 1.0 | 2.0 | 15.0 |
| **advance_planner \| couple** | 11110.0 | 1.454545 | 1.038746 | 1.0 | 1.0 | 1.0 | 2.0 | 25.0 |
| **advance_planner \| solo** | 3303.0 | 1.392976 | 0.901227 | 1.0 | 1.0 | 1.0 | 2.0 | 13.0 |
| **moderate_planner \| small_group** | 4969.0 | 1.244918 | 0.608735 | 1.0 | 1.0 | 1.0 | 1.0 | 9.0 |
| **moderate_planner \| large_group** | 2213.0 | 1.241753 | 0.667531 | 1.0 | 1.0 | 1.0 | 1.0 | 9.0 |
| **moderate_planner \| solo** | 1711.0 | 1.233197 | 0.623119 | 1.0 | 1.0 | 1.0 | 1.0 | 8.0 |
| **moderate_planner \| couple** | 5510.0 | 1.231216 | 0.573739 | 1.0 | 1.0 | 1.0 | 1.0 | 10.0 |
| **short_planner \| solo** | 5215.0 | 1.215340 | 0.656205 | 1.0 | 1.0 | 1.0 | 1.0 | 25.0 |
| **short_planner \| large_group** | 6512.0 | 1.206388 | 0.575523 | 1.0 | 1.0 | 1.0 | 1.0 | 13.0 |
| **short_planner \| small_group** | 14505.0 | 1.203585 | 0.564254 | 1.0 | 1.0 | 1.0 | 1.0 | 20.0 |
| **short_planner \| couple** | 16857.0 | 1.199324 | 0.569154 | 1.0 | 1.0 | 1.0 | 1.0 | 21.0 |
| **same_day \| solo** | 27588.0 | 1.193852 | 1.613655 | 1.0 | 1.0 | 1.0 | 1.0 | 252.0 |
| **same_day \| couple** | 80919.0 | 1.184184 | 2.279568 | 1.0 | 1.0 | 1.0 | 1.0 | 625.0 |
| **next_day \| solo** | 11399.0 | 1.180893 | 0.555326 | 1.0 | 1.0 | 1.0 | 1.0 | 19.0 |
| **same_day \| small_group** | 71450.0 | 1.173030 | 1.881951 | 1.0 | 1.0 | 1.0 | 1.0 | 476.0 |
| **same_day \| large_group** | 28961.0 | 1.166949 | 2.430397 | 1.0 | 1.0 | 1.0 | 1.0 | 392.0 |
| **next_day \| couple** | 34325.0 | 1.164195 | 0.563645 | 1.0 | 1.0 | 1.0 | 1.0 | 32.0 |
| **next_day \| small_group** | 29352.0 | 1.145373 | 0.523402 | 1.0 | 1.0 | 1.0 | 1.0 | 26.0 |
| **next_day \| large_group** | 12431.0 | 1.139570 | 0.671381 | 1.0 | 1.0 | 1.0 | 1.0 | 34.0 |

In [81]:

```
lead_time_mean = (
    persona_cust
    .assign(
        lead_group=lambda x: x['core_persona'].str.split(' | ').str[0]
    )
    .groupby('lead_group')['booking_count']
    .mean()
    .sort_values(ascending=False)
)
print("Average booking counts per Lead Day group:")
lead_time_mean.round(2)
```

```
Average booking counts per Lead Day group:
```

Out[81]:
```
lead_group
advance_planner     1.45
moderate_planner    1.24
short_planner       1.20
same_day            1.18
next_day            1.16
Name: booking_count, dtype: float64
```

## Advance planners show higher avg booking count then all other categories (1.45).

## Moderate planner and short time planners have similar booking range(1.24 & 1.2) which is higher than urgent planners but lower than advance planners

## When it comes to urgent planners(same day and next day), solo travellers have higher avg booking count than other guedst ranges than other groups.

In [83]:
```
df_bookings.groupby('guest_size_bin')['customer_country'].nunique()
```

Out[83]:
```
guest_size_bin
couple         164
large_group    154
small_group    166
solo           157
Name: customer_country, dtype: int64
```

## Domestic vs International Traveller:

In [85]:
```
round(df_bookings['customer_country'].value_counts(normalize=True)*100,2)
```

Out[85]:
```
customer_country
United Arab Emirates    43.59
India                    7.63
United Kingdom           7.26
United States            5.26
Saudi Arabia             3.51
                        ...
Vanuatu                  0.00
Cape Verde               0.00
Timor-Leste              0.00
Equatorial Guinea        0.00
Burundi                  0.00
Name: proportion, Length: 182, dtype: float64
```

In [86]:
```
df_bookings['country_type'] = np.where(df_bookings['customer_country'] ==
round(df_bookings['country_type'].value_counts(normalize=True)*100,2)
```

Out[86]:

```
country_type
international    56.41
domestic         43.59
Name: proportion, dtype: float64
```

In [87]:

```python
pd.crosstab(df_bookings['country_type'], df_bookings['lead_days_bin'], no
```

Out[87]:

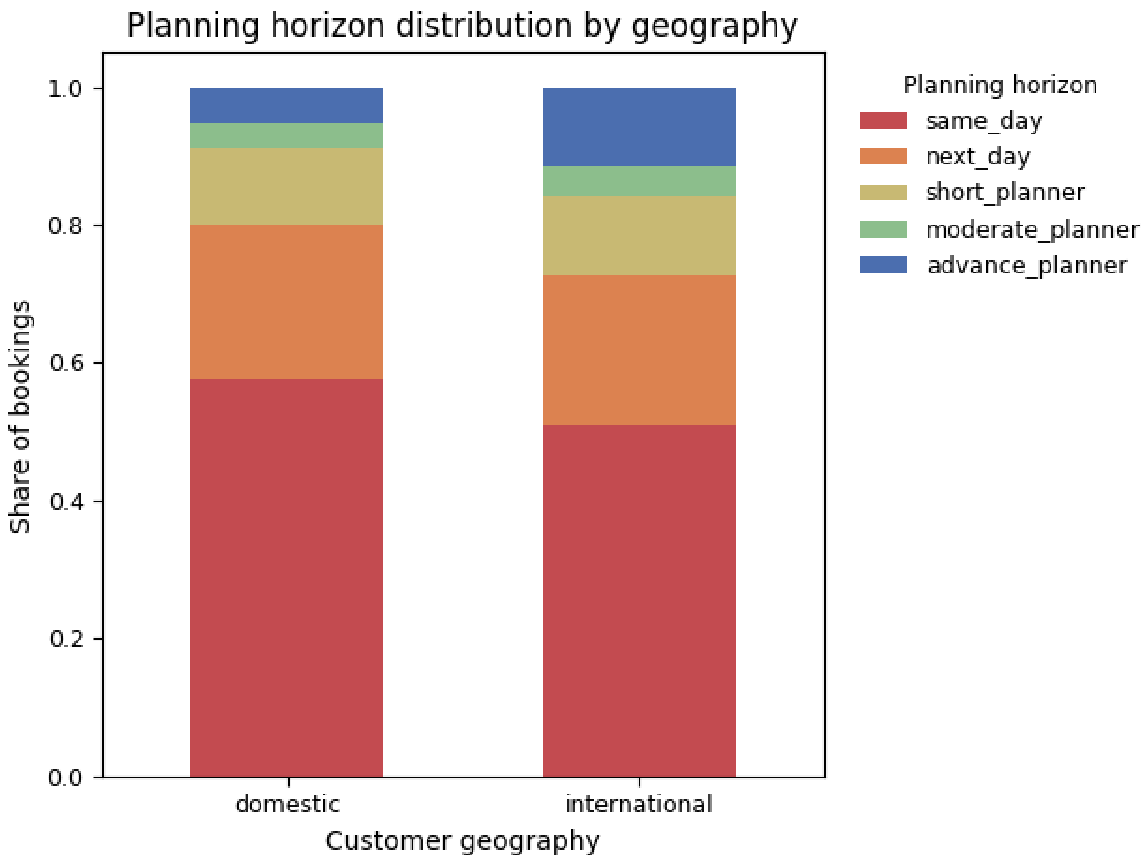
| lead_days_bin | advance_planner | moderate_planner | next_day | same_day | short_pl |
|---|---|---|---|---|---|
| **country_type** | | | | | |
| **domestic** | 0.05 | 0.03 | 0.23 | 0.58 | |
| **international** | 0.12 | 0.04 | 0.22 | 0.51 | |

In [88]:

```python
lead_country = pd.crosstab(
    df_bookings['country_type'],
    df_bookings['lead_days_bin'],
    normalize='index'
)

lead_country = lead_country[
    ['same_day', 'next_day', 'short_planner', 'moderate_planner', 'advanc
]

colors = [
    '#c44e52',
    '#dd8452',
    '#ccb974',
    '#8cbe8c',
    '#4c72b0'
]

# Plot
plt.figure(figsize=(6.5, 4))
lead_country.plot(
    kind='bar',
    stacked=True,
    width=0.55,
    color=colors
)

plt.ylabel('Share of bookings', fontsize=10)
plt.xlabel('Customer geography', fontsize=10)
plt.title('Planning horizon distribution by geography', fontsize=12)

plt.xticks(rotation=0, fontsize=9)
plt.yticks(fontsize=9)

plt.legend(
    title='Planning horizon',
    bbox_to_anchor=(1.02, 1),
    loc='upper left',
    frameon=False,
    fontsize=9,
    title_fontsize=9
)
```

```
plt.tight_layout()
plt.show()
```

```
<Figure size 650x400 with 0 Axes>
```



## Insights:

1. Both domestic and international customers tend to book on the same day (>50%)

2. International planners show higher incline towards advance planning (12% vs 5%)

In [90]:
```
pd.crosstab(df_bookings['country_type'], df_bookings['guest_size_bin'], n
```

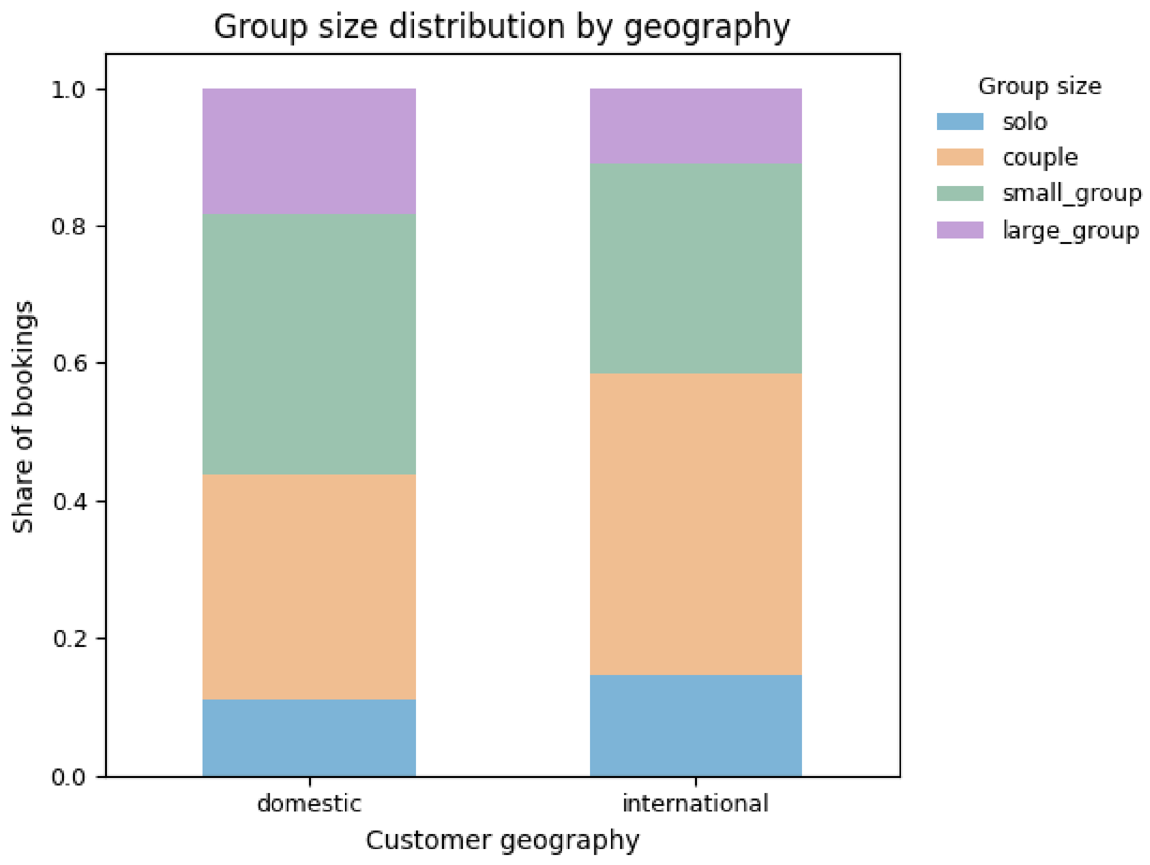
Out[90]:

| guest_size_bin | couple | large_group | small_group | solo |
|---|---|---|---|---|
| **country_type** | | | | |
| **domestic** | 0.33 | 0.18 | 0.38 | 0.11 |
| **international** | 0.44 | 0.11 | 0.31 | 0.15 |

In [91]:
```
guest_country = pd.crosstab(
    df_bookings['country_type'],
    df_bookings['guest_size_bin'],
    normalize='index'
)

guest_country = guest_country[
    ['solo', 'couple', 'small_group', 'large_group']
]
```

```
colors = [
    '#7fb7d8',
    '#f4c095',
    '#9ec7b3',
    '#c6a4d9'
]

plt.figure(figsize=(6.5, 4))
guest_country.plot(
    kind='bar',
    stacked=True,
    width=0.55,
    color=colors
)

plt.ylabel('Share of bookings', fontsize=10)
plt.xlabel('Customer geography', fontsize=10)
plt.title('Group size distribution by geography', fontsize=12)

plt.xticks(rotation=0, fontsize=9)
plt.yticks(fontsize=9)

plt.legend(
    title='Group size',
    bbox_to_anchor=(1.02, 1),
    loc='upper left',
    frameon=False,
    fontsize=9,
    title_fontsize=9
)

plt.tight_layout()
plt.show()
```

```
<Figure size 650x400 with 0 Axes>
```



### Insights:

International travellers skew more toward couples (44%), compared to domestic users (33%).

Domestic users show a higher combined share of group travel (small + large groups = 56%) compared to international users (42%).

Solo travel represents a slightly higher share among international users (15%) than domestic users (11%), though it remains a minority segment overall.

### Core Personas:

1. Lead Days

2. Group Size

### Secondary persona:

1. Booking frequency

## Observed differences across planning horizon, booking frequency, booking spacing behavior, and geographic distribution collectively validate the defined user personas as distinct and behaviorally meaningful segments

# Q2: Product Recommendations for User Personas:

## Customner level persona:

In [97]:
```
cust_persona = (
    df_bookings
    .groupby('customer_id', as_index=False)
    .agg(
        lead_days_bin=('lead_days_bin', lambda x: x.mode().iat[0] if len(
        guest_size_bin=('guest_size_bin', lambda x: x.mode().iat[0] if le
        observed_first_booking_date=('created_date', 'min'),
        booking_count=('booking_id', 'nunique')
    )
)

cust_persona['core_persona'] = cust_persona['lead_days_bin'] + ' | ' + cu

cust_persona
```

Out[97]:

| | customer_id | lead_days_bin | guest_size_bin | obser |
|---|---|---|---|---|
| **0** | cus_+++2FLNoNNnNp3D4uV368Q== | same_day | couple | 202 |
| **1** | cus_+++l8y+TeJMnyFczrcgx/g== | advance_planner | couple | 202 |
| **2** | cus_+++KHWAC7nsBI2iqk4WAYw== | next_day | large_group | 202 |
| **3** | cus_+++qMCXtvTDun8bkp7wGgw== | same_day | small_group | 202 |
| **4** | cus_++0ldLeirvA7FSulur1uBw== | next_day | couple | 202 |
| **...** | ... | ... | ... | |
| **318962** | cus_zzva2YoVsdsGJIFxw0qS4g== | same_day | solo | 202 |
| **318963** | cus_zzwbTtkjWFZPy2weJocHAQ== | short_planner | small_group | 202 |
| **318964** | cus_zzwcM+J+uWG56ppowu+V2w== | same_day | solo | 20 |
| **318965** | cus_zzy5DVD6m16y8Ajn9qQwWg== | same_day | couple | 202 |
| **318966** | cus_zzygHAYXL/X1fa+ZRjD++Q== | same_day | small_group | 202 |

318967 rows × 6 columns

In [98]:
```
cust_persona['core_persona'].value_counts()
```

Out[98]:
```
core_persona
same_day | couple                   73192
same_day | small_group              59805
next_day | couple                   31749
same_day | large_group              25028
next_day | small_group              24521
same_day | solo                     20599
short_planner | couple              11670
next_day | large_group              10971
advance_planner | couple            10597
short_planner | small_group          9436
next_day | solo                      8779
advance_planner | small_group        7803
moderate_planner | couple            4619
short_planner | large_group          4368
advance_planner | large_group        3670
moderate_planner | small_group       3633
short_planner | solo                 3295
advance_planner | solo               2390
moderate_planner | large_group       1699
moderate_planner | solo              1143
Name: count, dtype: int64
```

In [99]:
```
df_events_persona = df_events.merge(cust_persona, on='customer_id', how='
df_events_persona
```

Out[99]:

| | event_id | event_timestamp | event_na |
|---|---|---|---|
| **0** | evt_347e97a21c984d3790e0a52ebf0c2f99 | 2022-09-24 07:40:50+00:00 | product_page_v |
| **1** | evt_19f165ad9c1b4943b037652bce725072 | 2022-09-24 08:06:01+00:00 | product_page_v |
| **2** | evt_6daf55c2a5f24b0a9f058f35ac244e5b | 2022-12-24 05:35:26+00:00 | product_page_v |
| **3** | evt_77e141a5899a47caa4201a5da17a7d58 | 2022-09-25 12:58:14+00:00 | product_page_v |
| **4** | evt_8e5b21f606f1499f9cbace8f2141681c | 2022-09-25 13:13:14+00:00 | product_page_v |
| **...** | ... | ... | |
| **267712** | evt_0e20ad5fc94f4c4d96431aed7460698c | 2022-12-09 13:36:09+00:00 | product_page_v |
| **267713** | evt_9c83dd6c202f4d6db06b74d1b1da38cf | 2022-12-07 08:02:07+00:00 | product_page_v |
| **267714** | evt_2b3789cd535f411db23d5633f5799089 | 2022-12-06 09:29:12+00:00 | product_page_v |
| **267715** | evt_6375be9b1673413c8b801520215f4fd0 | 2022-11-02 21:53:37+00:00 | product_page_v |
| **267716** | evt_296a14d12f4f4b0c8ba8ad615df7cfe0 | 2022-11-02 22:34:37+00:00 | product_page_v |

267717 rows × 37 columns

In [100…
```
df_events_persona.isna().sum()
```

Out[100…
```
event_id                             0
event_timestamp                      0
event_name                           0
customer_id                          0
session_id                           0
product_id                           0
booking_id                           0
event_type                           0
position                             0
page_type                            0
device                               0
is_logged_in                         0
smart_rec_enabled                    0
rec_impressions                      0
rec_clicks                           0
rec_slot_position                    0
search_filters_used                  0
num_filters_applied                  0
search_query_length                  0
web_session_traffic_origin_country   0
web_session_traffic_source           0
web_session_traffic_medium           0
web_session_campaign_name            0
city                                 0
is_first_time_customer               0
customer_lifetime_bookings           0
experiment_bucket                    0
active_discount_flag                 0
supply_tier                          0
event_date                           0
event_time                           0
event_hour                           0
lead_days_bin                        0
guest_size_bin                       0
observed_first_booking_date          0
booking_count                        0
core_persona                         0
dtype: int64
```

## First-booking session number dist:

In [102…
```
session_start = (
    df_events_persona
    .groupby(['customer_id','session_id'])['event_timestamp']
    .min()
    .reset_index(name='session_start_time')
)

session_start['session_number'] = (
    session_start
    .sort_values('session_start_time')
    .groupby('customer_id')
    .cumcount() + 1
)

booking_sessions = (
    df_events_persona[df_events_persona['booking_id'].notna()]
    [['customer_id','session_id']]
    .drop_duplicates()
)

booking_sessions = booking_sessions.merge(
    session_start,
    on=['customer_id','session_id'],
    how='left'
)

round(booking_sessions['session_number'].value_counts(normalize=True)*100
```

Out[102…

```
session_number
1      69.2
2      18.4
3       6.1
4       2.6
5       1.3
       ...
342     0.0
341     0.0
340     0.0
339     0.0
481     0.0
Name: proportion, Length: 524, dtype: float64
```

In [103…

```
booking_sessions = booking_sessions.merge(
    cust_persona[['customer_id','lead_days_bin']],
    on='customer_id',
    how='left'
)

round(
    booking_sessions
    .groupby('lead_days_bin')['session_number']
    .value_counts(normalize=True)
    .mul(100)
    .unstack()
    .fillna(0),
    1
)
```

Out[103…

| session_number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| lead_days_bin | | | | | | | | | | | | | |
| advance_planner | 50.7 | 23.9 | 10.8 | 5.9 | 3.4 | 2.0 | 1.2 | 0.7 | 0.4 | 0.3 | 0.2 | 0.1 | 0.1 |
| moderate_planner | 57.2 | 24.4 | 9.1 | 4.2 | 2.2 | 1.2 | 0.7 | 0.4 | 0.2 | 0.1 | 0.1 | 0.1 | 0.0 |
| next_day | 69.1 | 21.2 | 5.4 | 2.2 | 0.9 | 0.4 | 0.2 | 0.1 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 |
| same_day | 75.7 | 15.2 | 4.5 | 1.7 | 0.8 | 0.4 | 0.2 | 0.1 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 |
| short_planner | 68.0 | 17.9 | 7.7 | 3.1 | 1.6 | 0.9 | 0.4 | 0.2 | 0.1 | 0.1 | 0.0 | 0.0 | 0.0 |

## Key observations

- ~70% of users book in their first session

- ~88% book within first two sessions
- Clear gradient by planning horizon:
- same_day users are most single-session
- advance planners show meaningful multi-session behavior
- Planning horizon strongly correlates with decision structure

## Search usage by persona

In [106…

```
# Sessions where booking happened:
booking_sessions_simple = (df_events_persona[df_events_persona['booking_i

# Keep only events from booking sessions
events_booking_session = df_events_persona.merge(booking_sessions_simple,

events_booking_session
```

Out[106…

| | event_id | event_timestamp | event_na |
|---|---|---|---|
| **0** | evt_77e141a5899a47caa4201a5da17a7d58 | 2022-09-25 12:58:14+00:00 | product_page_ |
| **1** | evt_8e5b21f606f1499f9cbace8f2141681c | 2022-09-25 13:13:14+00:00 | product_page_ |
| **2** | evt_55e565f8aae14c8f946b7629cdbc0d7d | 2022-12-21 11:57:18+00:00 | product_page_ |
| **3** | evt_6af91b75acd34f9c8d77108319901337 | 2022-11-28 05:07:55+00:00 | product_page_ |
| **4** | evt_61202dfac13c488784b591dad46dc783 | 2022-12-01 16:34:45+00:00 | product_page_ |
| **...** | ... | ... | |
| **156519** | evt_0de9e4dee8a1471a9dfe5e88fef7fd0c | 2022-09-25 09:01:22+00:00 | product_page_ |
| **156520** | evt_73d8bea12d5c4eb8af7ed82788a37785 | 2022-11-07 08:08:03+00:00 | product_page_ |
| **156521** | evt_587b9d2897294ef0920d1156079549bd | 2022-10-14 18:07:36+00:00 | product_page_ |
| **156522** | evt_12c10e79c2f14fa6959b2b8b87aa60d6 | 2022-11-06 09:16:44+00:00 | product_page_ |
| **156523** | evt_546015b81cb44eb1b22b6993e951c9bb | 2022-11-02 06:46:02+00:00 | product_page_ |

156524 rows × 37 columns

In [107…

```
events_booking_session.groupby('core_persona')['customer_id'].nunique().r
```

Out[107…

| | core_persona | count |
|---|---|---|
| 12 | same_day \| couple | 10568 |
| 14 | same_day \| small_group | 8892 |
| 8 | next_day \| couple | 4554 |
| 13 | same_day \| large_group | 3790 |
| 10 | next_day \| small_group | 3485 |
| 15 | same_day \| solo | 3256 |
| 0 | advance_planner \| couple | 2057 |
| 16 | short_planner \| couple | 1700 |
| 9 | next_day \| large_group | 1658 |
| 2 | advance_planner \| small_group | 1565 |
| 11 | next_day \| solo | 1498 |
| 18 | short_planner \| small_group | 1457 |
| 4 | moderate_planner \| couple | 822 |
| 1 | advance_planner \| large_group | 780 |
| 6 | moderate_planner \| small_group | 699 |
| 17 | short_planner \| large_group | 666 |
| 19 | short_planner \| solo | 584 |
| 3 | advance_planner \| solo | 481 |
| 5 | moderate_planner \| large_group | 324 |
| 7 | moderate_planner \| solo | 216 |

In [108…

```
search_usage = (
    events_booking_session
    .groupby(['customer_id','core_persona'])['search_filters_used']
    .max()
    .reset_index()
)

round(search_usage.groupby('core_persona')['search_filters_used'].mean(),
```

Out[108…

| | core_persona | mean |
|---|---|---|
| 2 | advance_planner \| small_group | 0.86 |
| 0 | advance_planner \| couple | 0.84 |
| 1 | advance_planner \| large_group | 0.84 |
| 5 | moderate_planner \| large_group | 0.83 |
| 6 | moderate_planner \| small_group | 0.83 |
| 17 | short_planner \| large_group | 0.83 |
| 3 | advance_planner \| solo | 0.82 |
| 4 | moderate_planner \| couple | 0.82 |
| 7 | moderate_planner \| solo | 0.82 |
| 18 | short_planner \| small_group | 0.80 |
| 16 | short_planner \| couple | 0.79 |
| 10 | next_day \| small_group | 0.79 |
| 8 | next_day \| couple | 0.79 |
| 11 | next_day \| solo | 0.78 |
| 9 | next_day \| large_group | 0.78 |
| 19 | short_planner \| solo | 0.78 |
| 12 | same_day \| couple | 0.77 |
| 13 | same_day \| large_group | 0.77 |
| 15 | same_day \| solo | 0.77 |
| 14 | same_day \| small_group | 0.76 |

In [109…

```
filters_usage = (
    events_booking_session
    .groupby(['customer_id','core_persona'])['num_filters_applied']
    .max()
    .reset_index()
)

round(filters_usage.groupby('core_persona')['num_filters_applied'].mean()
```

Out[109…
```
core_persona
advance_planner | couple          2.3
advance_planner | large_group     2.3
advance_planner | small_group     2.2
advance_planner | solo            2.2
moderate_planner | couple         2.3
moderate_planner | large_group    2.2
moderate_planner | small_group    2.2
moderate_planner | solo           2.2
next_day | couple                 2.1
next_day | large_group            2.1
next_day | small_group            2.1
next_day | solo                   2.1
same_day | couple                 2.1
same_day | large_group            2.0
same_day | small_group            2.0
same_day | solo                   2.1
short_planner | couple            2.1
short_planner | large_group       2.1
short_planner | small_group       2.1
short_planner | solo              2.0
Name: num_filters_applied, dtype: float64
```

No significant difference in pattern across personas

## First Time VS Returning user by persona:

In [112…
```
df_events.groupby('is_first_time_customer')['customer_id'].nunique()
```

Out[112…
```
is_first_time_customer
0     35621
1    108384
Name: customer_id, dtype: int64
```

In [113…
```
first_time_by_persona = (
    events_booking_session
    .groupby(['customer_id','core_persona'])['is_first_time_customer']
    .max()
    .reset_index()
)

round(first_time_by_persona.groupby('core_persona')['is_first_time_custom
```

Out[113…

| | core_persona | mean |
|---|---|---|
| 19 | short_planner \| solo | 98.3 |
| 15 | same_day \| solo | 98.2 |
| 11 | next_day \| solo | 97.1 |
| 3 | advance_planner \| solo | 97.1 |
| 17 | short_planner \| large_group | 97.0 |
| 18 | short_planner \| small_group | 96.7 |
| 16 | short_planner \| couple | 96.6 |
| 7 | moderate_planner \| solo | 95.8 |
| 2 | advance_planner \| small_group | 94.9 |
| 14 | same_day \| small_group | 94.4 |
| 6 | moderate_planner \| small_group | 94.1 |
| 10 | next_day \| small_group | 93.4 |
| 12 | same_day \| couple | 93.1 |
| 0 | advance_planner \| couple | 93.0 |
| 13 | same_day \| large_group | 91.9 |
| 8 | next_day \| couple | 91.9 |
| 4 | moderate_planner \| couple | 91.6 |
| 1 | advance_planner \| large_group | 91.3 |
| 5 | moderate_planner \| large_group | 90.1 |
| 9 | next_day \| large_group | 90.0 |

In [114…

```
first_time_by_group = (
    events_booking_session
    .groupby(['customer_id','guest_size_bin'])['is_first_time_customer']
    .max()
    .reset_index()
)

round(first_time_by_group.groupby('guest_size_bin')['is_first_time_custom
```

Out[114…

```
guest_size_bin
couple          93.1
large_group     91.8
small_group     94.4
solo            97.7
Name: is_first_time_customer, dtype: float64
```

## Bookings are largely dominated by first timers (>90% across all personas)

## Solo travellers seems to skew more towards first time travellers

## Page/Product View:

In [117…

```
exploration_depth = (
    events_booking_session
    .query("event_name == 'product_page_view'")
    .groupby(['customer_id','core_persona'])
    .size()
    .reset_index(name='product_page_views')
)

round(
    exploration_depth
    .groupby('core_persona')['product_page_views']
    .mean(),
    2
)
```

Out[117…

```
core_persona
advance_planner | couple          1.90
advance_planner | large_group     2.10
advance_planner | small_group     2.07
advance_planner | solo            1.85
moderate_planner | couple         1.75
moderate_planner | large_group    1.88
moderate_planner | small_group    1.74
moderate_planner | solo           1.65
next_day | couple                 1.47
next_day | large_group            1.45
next_day | small_group            1.45
next_day | solo                   1.41
same_day | couple                 1.38
same_day | large_group            1.42
same_day | small_group            1.29
same_day | solo                   1.29
short_planner | couple            1.46
short_planner | large_group       1.58
short_planner | small_group       1.48
short_planner | solo              1.36
Name: product_page_views, dtype: float64
```

In [118…

```
exploration_depth = (
    events_booking_session
    .query("event_name == 'product_page_view'")
    .groupby(['customer_id','guest_size_bin'])
    .size()
    .reset_index(name='product_page_views')
)

round(
    exploration_depth
    .groupby('guest_size_bin')['product_page_views']
    .mean(),
    2
)
```

Out[118…

```
guest_size_bin
couple         1.48
large_group    1.54
small_group    1.43
solo           1.38
Name: product_page_views, dtype: float64
```

In [119…

```
exploration_depth = (
    events_booking_session
    .groupby(['customer_id','core_persona'])
    .size()
    .reset_index(name='total_page_views')
)

round(
    exploration_depth
    .groupby('core_persona')['total_page_views']
    .mean(),
    1
)
```

Out[119…

```
core_persona
advance_planner | couple          4.1
advance_planner | large_group     4.6
advance_planner | small_group     4.5
advance_planner | solo            4.0
moderate_planner | couple         3.8
moderate_planner | large_group    4.1
moderate_planner | small_group    3.8
moderate_planner | solo           3.7
next_day | couple                 3.2
next_day | large_group            3.2
next_day | small_group            3.2
next_day | solo                   3.1
same_day | couple                 3.0
same_day | large_group            3.1
same_day | small_group            2.8
same_day | solo                   2.8
short_planner | couple            3.2
short_planner | large_group       3.5
short_planner | small_group       3.2
short_planner | solo              2.9
Name: total_page_views, dtype: float64
```

Product exploration increases with increase in planning horizon

Larger groups → more comparison

Solo users → fastest decision-makers

## Recommendation exposure & interaction (impressions / clicks)

In [122…

```
rec_exposure = (
    events_booking_session
    .groupby(['customer_id','core_persona'])['rec_impressions']
    .max()
    .reset_index()
)

round(
    (rec_exposure['rec_impressions'] > 0)
    .groupby(rec_exposure['core_persona'])
    .mean() * 100,
    1
).reset_index(name='mean').sort_values(by='mean', ascending=False)
```

Out[122…

| | core_persona | mean |
|---|---|---|
| 7 | moderate_planner \| solo | 60.6 |
| 5 | moderate_planner \| large_group | 57.4 |
| 16 | short_planner \| couple | 56.0 |
| 9 | next_day \| large_group | 54.8 |
| 10 | next_day \| small_group | 54.3 |
| 11 | next_day \| solo | 54.3 |
| 17 | short_planner \| large_group | 53.9 |
| 0 | advance_planner \| couple | 53.4 |
| 15 | same_day \| solo | 53.0 |
| 1 | advance_planner \| large_group | 52.9 |
| 13 | same_day \| large_group | 52.8 |
| 3 | advance_planner \| solo | 52.6 |
| 12 | same_day \| couple | 52.5 |
| 14 | same_day \| small_group | 52.5 |
| 6 | moderate_planner \| small_group | 52.4 |
| 4 | moderate_planner \| couple | 52.4 |
| 2 | advance_planner \| small_group | 52.1 |
| 8 | next_day \| couple | 51.2 |
| 18 | short_planner \| small_group | 50.7 |
| 19 | short_planner \| solo | 49.8 |

In [123…

```
rec_metrics = (
    events_booking_session
    .groupby(['customer_id','core_persona'])[['rec_impressions','rec_clic
    .sum()
    .reset_index()
)

rec_ctr = (
    rec_metrics
    .groupby('core_persona')
    .apply(lambda x: x['rec_clicks'].sum() / x['rec_impressions'].sum())
)
```

```
round(rec_ctr, 3)
```

Out[123…

```
core_persona
advance_planner | couple         0.068
advance_planner | large_group    0.074
advance_planner | small_group    0.065
advance_planner | solo           0.061
moderate_planner | couple        0.067
moderate_planner | large_group   0.068
moderate_planner | small_group   0.074
moderate_planner | solo          0.071
next_day | couple                0.067
next_day | large_group           0.069
next_day | small_group           0.071
next_day | solo                  0.067
same_day | couple                0.072
same_day | large_group           0.071
same_day | small_group           0.069
same_day | solo                  0.067
short_planner | couple           0.071
short_planner | large_group      0.067
short_planner | small_group      0.072
short_planner | solo             0.069
dtype: float64
```

No significant pattern

## Q3: Smart Recommendations Evaluation

In [126…

```
events_scope = df_events_persona[
    (df_events_persona["city"].str.lower() == "dubai") &
    (df_events_persona["event_timestamp"].dt.date >= pd.to_datetime("2022
    (df_events_persona["event_timestamp"].dt.date <= pd.to_datetime("2022
].copy()

df_sess = (
    events_scope
    .groupby(["session_id", "customer_id"], as_index=False)
    .agg(
        variant=("experiment_bucket", "first"),
        rec_impressions=("rec_impressions", "sum"),
        rec_clicks=("rec_clicks", "sum"),
        converted=("booking_id", lambda s: int(((s.notna()) & (s != "no_b
        is_first_time=("is_first_time_customer", "max"),
        device=("device", "first"),
        traffic_source=("web_session_traffic_source", "first"),
        traffic_medium=("web_session_traffic_medium", "first"),
        origin_country=("web_session_traffic_origin_country", "first"),
        lead_days_bin=("lead_days_bin", lambda x: x.mode().iat[0] if len(
        guest_size_bin=("guest_size_bin", lambda x: x.mode().iat[0] if le
        core_persona=("core_persona", lambda x: x.mode().iat[0] if len(x.
    )
)

df_sess["variant"] = df_sess["variant"].astype(str).str.lower()
df_sess = df_sess[df_sess["variant"].isin(["control", "treatment"])].copy
```

```
df_sess["rec_exposed"] = (df_sess["rec_impressions"] > 0).astype(int)
df_sess["ctr"] = df_sess["rec_clicks"] / df_sess["rec_impressions"].repla

df_sess.head()
```

Out[126…

| | session_id | customer_id | vari |
|---|---|---|---|
| **0** | sess_bid_++6x3IbS1bjKjo4DnYH8ZQ== | cus_O0fB1caruQtiAryYRDVbgg== | treatm |
| **1** | sess_bid_++OeArHaM2mn17jNwu5tVQ== | cus_Pgb33RLKpl+PF902ceK3Ww== | con |
| **2** | sess_bid_++QirfTO52og/YfvDw0YuQ== | cus_2IhzJzRo+Ma56vtl2iGcLA== | con |
| **3** | sess_bid_++lFIEzhddPjtypr6WhzFg== | cus_h4QzJNALTorDxgmRH12kyw== | con |
| **4** | sess_bid_++muqnHFXcxNPmCCVZBHjg== | cus_mCJJ6g5u9GBKEj07QoKDiQ== | treatm |

In [127…

```
def segment_metrics(df, segment_col, min_sessions=200):

    g = (
        df.groupby([segment_col, "variant"], dropna=False)
          .agg(
              sessions=("session_id", "nunique"),
              users=("customer_id", "nunique"),
              cvr=("converted", "mean"),
              clicks=("rec_clicks", "sum"),
              impr=("rec_impressions", "sum"),
          )
          .reset_index()
    )

    g["ctr"] = g["clicks"] / g["impr"].replace(0, np.nan)

    p = g.pivot(index=segment_col, columns="variant")

    out = pd.DataFrame(index=p.index)

    out["sessions_control"] = p[("sessions", "control")]
    out["sessions_treatment"] = p[("sessions", "treatment")]

    out["cvr_control"] = p[("cvr", "control")]
    out["cvr_treatment"] = p[("cvr", "treatment")]
    out["cvr_uplift_pp"] = (out["cvr_treatment"] - out["cvr_control"]) *

    out["ctr_treatment"] = p[("ctr", "treatment")]

    total_sessions = out["sessions_control"].fillna(0) + out["sessions_tr
    out = out[total_sessions >= min_sessions].copy()

    for col in ["cvr_control", "cvr_treatment", "cvr_uplift_pp", "ctr_tre
        out[col] = out[col].round(2)

    out["sessions_control"] = out["sessions_control"].astype(int)
    out["sessions_treatment"] = out["sessions_treatment"].astype(int)
    return out.reset_index().sort_values("cvr_uplift_pp", ascending=False
```

In [128…

```
lead_report = segment_metrics(df_sess, "lead_days_bin", min_sessions=300)
lead_report
```

Out[128…

| | lead_days_bin | sessions_control | sessions_treatment | cvr_control | cvr_treatm |
|---|---|---|---|---|---|
| **1** | moderate_planner | 570 | 508 | 0.39 | 0 |
| **2** | next_day | 2067 | 2014 | 0.41 | 0 |
| **3** | same_day | 4480 | 4476 | 0.41 | 0 |
| **4** | short_planner | 820 | 904 | 0.39 | 0 |
| **0** | advance_planner | 882 | 793 | 0.41 | 0 |

In [129…

```
group_report = segment_metrics(df_sess, "guest_size_bin", min_sessions=30
group_report
```

Out[129…

| | guest_size_bin | sessions_control | sessions_treatment | cvr_control | cvr_treatmen |
|---|---|---|---|---|---|
| **1** | large_group | 1578 | 1535 | 0.40 | 0.4 |
| **3** | solo | 1023 | 987 | 0.38 | 0.4 |
| **2** | small_group | 2927 | 2924 | 0.41 | 0.4 |
| **0** | couple | 3291 | 3249 | 0.41 | 0.4 |

In [130…

```
core_report = segment_metrics(df_sess, "core_persona", min_sessions=500)
core_report
```

Out[130…

| | core_persona | sessions_control | sessions_treatment | cvr_control | cvr_treatm |
|---|---|---|---|---|---|
| **9** | short_planner \| couple | 273 | 293 | 0.38 | 0 |
| **8** | same_day \| solo | 555 | 524 | 0.39 | 0 |
| **4** | next_day \| small_group | 645 | 666 | 0.40 | 0 |
| **6** | same_day \| large_group | 730 | 757 | 0.39 | 0 |
| **10** | short_planner \| small_group | 251 | 293 | 0.36 | 0 |
| **3** | next_day \| large_group | 406 | 378 | 0.38 | 0 |
| **1** | advance_planner \| small_group | 288 | 267 | 0.43 | 0 |
| **0** | advance_planner \| couple | 358 | 340 | 0.40 | 0 |
| **7** | same_day \| small_group | 1537 | 1508 | 0.41 | 0 |
| **5** | same_day \| couple | 1658 | 1687 | 0.42 | 0 |
| **2** | next_day \| couple | 789 | 751 | 0.42 | 0 |

In [131…

```
first_time_report = segment_metrics(df_sess, "is_first_time", min_session
first_time_report
```

Out[131…

| | is_first_time | sessions_control | sessions_treatment | cvr_control | cvr_treatment |
|---|---|---|---|---|---|
| **1** | 1 | 5760 | 5705 | 0.40 | 0.45 |
| **0** | 0 | 3059 | 2990 | 0.42 | 0.47 |

In [132…

```
device_report = segment_metrics(df_sess, "device", min_sessions=500)
device_report
```

Out[132…

| | device | sessions_control | sessions_treatment | cvr_control | cvr_treatment | cvr_ |
|---|---|---|---|---|---|---|
| **2** | iOS App | 550 | 435 | 0.44 | 0.51 | |
| **1** | Mobile | 5984 | 6046 | 0.40 | 0.45 | |
| **0** | Desktop | 2071 | 1908 | 0.41 | 0.45 | |

In [133…

```
source_report = segment_metrics(df_sess, "traffic_source", min_sessions=8
source_report
```

Out[133…

| | traffic_source | sessions_control | sessions_treatment | cvr_control | cvr_tre |
|---|---|---|---|---|---|
| **0** | (direct) | 750 | 759 | 0.38 | |
| **2** | unkown/no_web_traffic | 1069 | 1045 | 0.42 | |
| **1** | google | 6471 | 6399 | 0.41 | |

In [134…

```
medium_report = segment_metrics(df_sess, "traffic_medium", min_sessions=8
medium_report
```

Out[134…

| | traffic_medium | sessions_control | sessions_treatment | cvr_control | cvr_tre |
|---|---|---|---|---|---|
| **0** | (none) | 750 | 759 | 0.38 | |
| **1** | cpc | 5401 | 5370 | 0.40 | |
| **3** | unkown/no_web_traffic | 1069 | 1045 | 0.42 | |
| **2** | organic | 1158 | 1098 | 0.41 | |

### Significant positive impact of Smart Rec is visible across all segments:

In [234…

```
pip uninstall -y jupyter_contrib_nbextensions
```

```
Found existing installation: jupyter_contrib_nbextensions 0.7.0
Uninstalling jupyter_contrib_nbextensions-0.7.0:
  Successfully uninstalled jupyter_contrib_nbextensions-0.7.0
Note: you may need to restart the kernel to use updated packages.
```

In [240…

```
!jupyter nbconvert "Dubai Booking User Persona Defination.ipynb" --to htm
```

```
[NbConvertApp] Converting notebook Dubai Booking User Persona Defination.i
pynb to html
[NbConvertApp] WARNING | Alternative text is missing on 3 image(s).
[NbConvertApp] Writing 775522 bytes to Dubai Booking User Persona Definati
on.html
```

In [ ]: