Untitled

January 12, 2026

[35]:
```
# HOTEL BOOKING ANALYSIS - COMPLETE PYTHON CODE
# TravClan Business Analyst Assignment
```

[36]:
```
# STEP 1: IMPORT LIBRARIES
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
import warnings
warnings.filterwarnings('ignore')

# For statistical analysis
from scipy import stats
from scipy.stats import chi2_contingency
```

[37]:
```
# Set display options
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_rows', 100)

# Set plot style
plt.style.use('seaborn-v0_8-darkgrid')
sns.set_palette("husl")

print("HOTEL BOOKING ANALYSIS - TRAVCLAN ASSIGNMENT")
```

```
HOTEL BOOKING ANALYSIS - TRAVCLAN ASSIGNMENT
```

[38]:
```
# STEP 2: LOAD DATA
print("LOADING DATA")

# Load the cleaned CSV file
df = pd.read_csv('Excel1.csv')

print(f"Data loaded successfully!")
print(f"Total rows: {len(df):,}")
print(f"Total columns: {len(df.columns)}")
```

```
LOADING DATA
Data loaded successfully!
Total rows: 30,000
Total columns: 44
```

[39]:
```python
# STEP 3: DATA OVERVIEW
print("DATA OVERVIEW")

print("\nFirst 5 rows:")
print(df.head())

print("\nData Types:")
print(df.dtypes)

print("\nBasic Statistics:")
print(df.describe())

print("\nMissing Values:")
missing = df.isnull().sum()
missing_pct = (missing / len(df)) * 100
missing_df = pd.DataFrame({
    'Missing_Count': missing,
    'Percentage': missing_pct
})
print(missing_df[missing_df['Missing_Count'] > 0])
```

```
DATA OVERVIEW

First 5 rows:
   customer_id  property_id           city  star_rating booking_date  \
0          492            3  San Francisco            4   01-04-2024
1          180            3         Dallas            3   01-04-2024
2           50            5         Dallas            3   01-04-2024
3          294            3        Orlando            4   01-04-2024
4           40            5        Seattle            5   01-04-2024

  check_in_date check_out_date room_type  num_rooms_booked stay_type  \
0    24-05-2024     26-05-2024  Standard                 1   Leisure
1    10-05-2024     17-05-2024    Deluxe                 1   Leisure
2    31-05-2024     05-06-2024    Deluxe                 1  Business
3    18-04-2024     24-04-2024    Deluxe                 3   Leisure
4           NaN            NaN    Deluxe                 1   Leisure

  booking_channel  booking_value  costprice   markup  selling_price  \
0      Mobile App        19361.0      19361   5981.0        25342.0
1      Mobile App         6137.0       6137   1896.0         8033.0
2             Web        22702.0      22702   7013.0        29715.0
3             Web        34068.0      34068  10524.0        44592.0
```

```
4      Mobile App        12127.0      12127   3746.0        15873.0

  payment_method refund_status  refund_amount channel_of_booking  \
0         PayPal           Yes         369.65                Web
1  Bank Transfer           Yes         492.51                Web
2     Debit Card           Yes           0.00                iOS
3  Bank Transfer           Yes         545.54            Android
4     Debit Card           Yes         211.37                Web

  booking_status travel_date  cashback  coupon_redeem Coupon USed?  \
0      Confirmed  04-03-2024      5.37            0.0           No
1      Confirmed  19-07-2024      7.16            0.0           No
2      Confirmed  22-03-2024      0.00            0.0           No
3      Confirmed  24-11-2024      7.93           24.5          Yes
4      Cancelled  02-03-2024      0.00            0.0           No

   length_of_stay  is_weekend_checkin  has_complete_dates  actual_length_stay  \
0             2.0                True                   1                 2.0
1             7.0                True                   1                 7.0
2             5.0                True                   1                 5.0
3             6.0               False                   1                 6.0
4             NaN               False                   1                 NaN

   total_cost profit_margin  profit_margin_pct booking_month  \
0     25342.0    ? 5,981.00               31.0        Apr-24
1      8033.0    ? 1,896.00               31.0        Apr-24
2     29715.0    ? 7,013.00               31.0        Apr-24
3     44592.0   ? 10,524.00               31.0        Apr-24
4     15873.0    ? 3,746.00               31.0        Apr-24

   booking_month_num  booking_year  days_before_checkin  is_cancelled  \
0                  4          2024                 53.0             0
1                  4          2024                 39.0             0
2                  4          2024                 60.0             0
3                  4          2024                 17.0             0
4                  4          2024                  NaN             1

   is_refunded  has_coupon  net_revenue  flag_outliers  negative_margins  \
0            1           0     24972.35              0                 0
1            1           0      7540.49              0                 0
2            1           0     29715.00              0                 0
3            1           1     44046.46              0                 0
4            1           0     15661.63              0                 0

   future_dates  invalid_stay_length average_booking_value
0             0                    0             25080.53
1             0                    0    standard_devaition
2             0                    0             11587.37
```

```
3             0                    0                    NaN
4             0                    0                    NaN

Data Types:
customer_id                int64
property_id                int64
city                      object
star_rating                int64
booking_date              object
check_in_date             object
check_out_date            object
room_type                 object
num_rooms_booked           int64
stay_type                 object
booking_channel           object
booking_value            float64
costprice                  int64
markup                   float64
selling_price            float64
payment_method            object
refund_status             object
refund_amount            float64
channel_of_booking        object
booking_status            object
travel_date               object
cashback                 float64
coupon_redeem            float64
Coupon USed?              object
length_of_stay           float64
is_weekend_checkin          bool
has_complete_dates         int64
actual_length_stay       float64
total_cost               float64
profit_margin             object
profit_margin_pct        float64
booking_month             object
booking_month_num          int64
booking_year               int64
days_before_checkin      float64
is_cancelled               int64
is_refunded                int64
has_coupon                 int64
net_revenue              float64
flag_outliers              int64
negative_margins           int64
future_dates               int64
invalid_stay_length        int64
average_booking_value     object
```

```
dtype: object

Basic Statistics:
        customer_id   property_id   star_rating  num_rooms_booked  \
count  30000.000000  30000.000000  30000.000000      30000.000000
mean     249.721767      3.777533      3.602033          1.352333
std      145.484836      2.081471      0.860241          0.574403
min        1.000000      1.000000      2.000000          1.000000
25%      123.000000      1.000000      3.000000          1.000000
50%      248.000000      3.000000      4.000000          1.000000
75%      376.000000      5.000000      4.000000          2.000000
max      499.000000      7.000000      5.000000          3.000000

       booking_value     costprice       markup  selling_price  refund_amount  \
count   30000.000000  30000.000000  30000.00000   30000.000000   30000.000000
mean    25080.525454  22541.494100   7199.75008   29741.244180     482.841137
std     11587.369335   9259.520787   3021.50829   12234.339848    2226.071521
min      1279.770000   3822.000000   1181.00000    5003.000000       0.000000
25%     15922.712500  15088.750000   4797.00000   19890.187500       0.000000
50%     24736.145000  22731.000000   7235.00000   30000.000000     216.080000
75%     33570.000000  30427.000000   9665.00000   40119.250000     596.400000
max     67764.460000  38200.000000  17556.50000   53021.800000   49888.910000

           cashback  coupon_redeem  length_of_stay  has_complete_dates  \
count  30000.000000   30000.000000    24532.000000        30000.000000
mean       3.615018       4.134972        4.006848            0.817767
std        4.584495      10.846211        2.001231            0.386043
min        0.000000      -3.530000        1.000000            0.000000
25%        0.000000       0.000000        2.000000            1.000000
50%        0.940000       0.000000        4.000000            1.000000
75%        6.790000       0.000000        6.000000            1.000000
max       21.270000      89.640000        7.000000            1.000000

       actual_length_stay    total_cost  profit_margin_pct  booking_month_num  \
count        24532.000000  30000.000000            30000.0       30000.000000
mean             4.006848  32280.275534               31.0           6.340067
std              2.001231  13694.936170                0.0           3.383450
min              1.000000   2553.770000               31.0           1.000000
25%              2.000000  21558.802500               31.0           4.000000
50%              4.000000  31936.960000               31.0           6.000000
75%              6.000000  42449.807500               31.0           9.000000
max              7.000000  82172.010000               31.0          12.000000

       booking_year  days_before_checkin  is_cancelled   is_refunded  \
count  30000.000000         24532.000000  30000.000000  30000.000000
mean    2024.302033            30.378485      0.192633      0.772733
std        0.459147            17.374172      0.394374      0.419073
min     2024.000000             1.000000      0.000000      0.000000
```

```
25%     2024.000000            15.000000       0.000000       1.000000
50%     2024.000000            30.000000       0.000000       1.000000
75%     2025.000000            45.000000       0.000000       1.000000
max     2025.000000            60.000000       1.000000       1.000000

         has_coupon   net_revenue  flag_outliers  negative_margins  \
count  30000.000000  30000.000000   30000.000000           30000.0
mean       0.206433  29258.403043       0.000767               0.0
std        0.404752  12371.746304       0.027679               0.0
min        0.000000     32.680000       0.000000               0.0
25%        0.000000  19386.415000       0.000000               0.0
50%        0.000000  29530.700000       0.000000               0.0
75%        0.000000  39724.270000       0.000000               0.0
max        1.000000  53021.800000       1.000000               0.0

       future_dates  invalid_stay_length
count       30000.0              30000.0
mean            0.0                  0.0
std             0.0                  0.0
min             0.0                  0.0
25%             0.0                  0.0
50%             0.0                  0.0
75%             0.0                  0.0
max             0.0                  0.0

Missing Values:
                       Missing_Count  Percentage
check_in_date                   5468   18.226667
check_out_date                  5468   18.226667
length_of_stay                  5468   18.226667
actual_length_stay              5468   18.226667
days_before_checkin             5468   18.226667
average_booking_value          29997   99.990000
```

[40]:
```python
# STEP 4: DATA TYPE CONVERSIONS
print("CONVERTING DATA TYPES")

# Convert date columns
date_columns = ['booking_date', 'check_in_date', 'check_out_date',␣
 ↪'travel_date']
for col in date_columns:
    if col in df.columns:
        df[col] = pd.to_datetime(df[col], errors='coerce')
        print(f" Converted {col} to datetime")

# Convert categorical columns
categorical_columns = ['city', 'room_type', 'stay_type', 'booking_channel',
```

```
                        'payment_method', 'refund_status', 'booking_status']
for col in categorical_columns:
    if col in df.columns:
        df[col] = df[col].astype('category')
        print(f" Converted {col} to category")
```

```
CONVERTING DATA TYPES
 Converted booking_date to datetime
 Converted check_in_date to datetime
 Converted check_out_date to datetime
 Converted travel_date to datetime
 Converted city to category
 Converted room_type to category
 Converted stay_type to category
 Converted booking_channel to category
 Converted payment_method to category
 Converted refund_status to category
 Converted booking_status to category
```

[41]:
```
# STEP 5: KEY BUSINESS METRICS (ERROR-FREE)

# Ensure numeric safety (extra protection)
safe_cols = [
    'selling_price',
    'costprice',
    'profit_margin',
    'booking_value',
    'profit_margin_pct',
    'actual_length_stay'
]

for col in safe_cols:
    if col in df.columns:
        df[col] = pd.to_numeric(df[col], errors='coerce')

metrics = {
    'Total Bookings': len(df),
    'Total Revenue': df['selling_price'].sum(),
    'Total Cost': df['costprice'].sum(),
    'Total Profit': df['profit_margin'].sum(),
    'Average Booking Value': df['booking_value'].mean(),
    'Average Profit Margin %': df['profit_margin_pct'].mean(),
    'Cancellation Rate %': df['is_cancelled'].mean() * 100,
    'Refund Rate %': df['is_refunded'].mean() * 100,
    'Average Stay Length': df['actual_length_stay'].mean(),
    'Total Customers': df['customer_id'].nunique(),
    'Total Properties': df['property_id'].nunique()
```

```
}

print("\nKEY BUSINESS METRICS\n")

for metric, value in metrics.items():
    if 'Rate' in metric or '%' in metric:
        print(f"{metric:.<40} {value:.2f}%")
    elif metric in ['Total Bookings', 'Total Customers', 'Total Properties']:
        print(f"{metric:.<40} {int(value):,}")
    else:
        print(f"{metric:.<40} ${float(value):,.2f}")
```

```
KEY BUSINESS METRICS

Total Bookings… 30,000
Total Revenue… $892,237,325.39
Total Cost… $676,244,823.00
Total Profit… $0.00
Average Booking Value… $25,080.53
Average Profit Margin %… 31.00%
Cancellation Rate %… 19.26%
Refund Rate %… 77.27%
Average Stay Length… $4.01
Total Customers… 499
Total Properties… 4
```

[42]:
```
# STEP 6: ANALYSIS 1 - CANCELLATION ANALYSIS
print("ANALYSIS 1: CANCELLATION PATTERNS")

# Cancellation by Booking Channel
cancel_by_channel = df.groupby('booking_channel').agg({
    'is_cancelled': ['mean', 'sum', 'count']
}).round(4)
cancel_by_channel.columns = ['Cancellation_Rate', 'Total_Cancelled',␣
 ↪'Total_Bookings']
cancel_by_channel['Cancellation_Rate'] = cancel_by_channel['Cancellation_Rate']␣
 ↪* 100
cancel_by_channel = cancel_by_channel.sort_values('Cancellation_Rate',␣
 ↪ascending=False)

print("\n Cancellation Rate by Booking Channel:")
print(cancel_by_channel)

# Cancellation by Room Type
cancel_by_room = df.groupby('room_type').agg({
    'is_cancelled': ['mean', 'sum', 'count']
```

```
}).round(4)
cancel_by_room.columns = ['Cancellation_Rate', 'Total_Cancelled',␣
 ↪'Total_Bookings']
cancel_by_room['Cancellation_Rate'] = cancel_by_room['Cancellation_Rate'] * 100
cancel_by_room = cancel_by_room.sort_values('Cancellation_Rate',␣
 ↪ascending=False)

print("\n Cancellation Rate by Room Type:")
print(cancel_by_room)

# Cancellation by Star Rating
cancel_by_star = df.groupby('star_rating').agg({
    'is_cancelled': ['mean', 'sum', 'count']
}).round(4)
cancel_by_star.columns = ['Cancellation_Rate', 'Total_Cancelled',␣
 ↪'Total_Bookings']
cancel_by_star['Cancellation_Rate'] = cancel_by_star['Cancellation_Rate'] * 100
cancel_by_star = cancel_by_star.sort_values('Cancellation_Rate',␣
 ↪ascending=False)

print("\n Cancellation Rate by Star Rating:")
print(cancel_by_star)

# Weekend vs Weekday Cancellation
cancel_by_weekend = df.groupby('is_weekend_checkin').agg({
    'is_cancelled': ['mean', 'sum', 'count']
}).round(4)
cancel_by_weekend.columns = ['Cancellation_Rate', 'Total_Cancelled',␣
 ↪'Total_Bookings']
cancel_by_weekend['Cancellation_Rate'] = cancel_by_weekend['Cancellation_Rate']␣
 ↪* 100

print("\n Cancellation Rate: Weekend vs Weekday:")
print(cancel_by_weekend)
```

```
ANALYSIS 1: CANCELLATION PATTERNS

 Cancellation Rate by Booking Channel:
                 Cancellation_Rate  Total_Cancelled  Total_Bookings
booking_channel
Travel Agent                 26.29              786            2990
Mobile App                   22.42             2693           12009
Web                          15.33             2300           15001

 Cancellation Rate by Room Type:
            Cancellation_Rate  Total_Cancelled  Total_Bookings
room_type
```

```
Standard                22.12             3662           16552
Suite                   15.89              472            2970
Deluxe                  15.70             1645           10478

 Cancellation Rate by Star Rating:
             Cancellation_Rate  Total_Cancelled  Total_Bookings
star_rating
3                        20.42             2136           10460
2                        19.97              598            2995
5                        19.11              862            4511
4                        18.14             2183           12034

 Cancellation Rate: Weekend vs Weekday:
                    Cancellation_Rate  Total_Cancelled  Total_Bookings
is_weekend_checkin
False                           27.20             5312           19529
True                             4.46              467           10471
```

[43]:
```
# STEP 7: ANALYSIS 2 - REVENUE ANALYSIS
print("ANALYSIS 2: REVENUE PERFORMANCE")

# Revenue by Booking Channel
revenue_by_channel = df.groupby('booking_channel').agg({
    'selling_price': ['sum', 'mean', 'count'],
    'profit_margin': ['sum', 'mean'],
    'is_cancelled': 'mean'
}).round(2)
revenue_by_channel.columns = ['Total_Revenue', 'Avg_Revenue', 'Bookings',
                              'Total_Profit', 'Avg_Profit', 'Cancel_Rate']
revenue_by_channel['Cancel_Rate'] = revenue_by_channel['Cancel_Rate'] * 100
revenue_by_channel = revenue_by_channel.sort_values('Total_Revenue',␣
 ↪ascending=False)

print("\n Revenue Performance by Booking Channel:")
print(revenue_by_channel)

# Revenue by City
revenue_by_city = df.groupby('city').agg({
    'selling_price': ['sum', 'mean', 'count'],
    'profit_margin': ['sum', 'mean']
}).round(2)
revenue_by_city.columns = ['Total_Revenue', 'Avg_Revenue', 'Bookings',
                           'Total_Profit', 'Avg_Profit']
revenue_by_city = revenue_by_city.sort_values('Total_Revenue', ascending=False).
 ↪head(10)

print("\n Top 10 Cities by Revenue:")
```

```python
print(revenue_by_city)

# Revenue by Star Rating
revenue_by_star = df.groupby('star_rating').agg({
    'selling_price': ['sum', 'mean', 'count'],
    'profit_margin': ['sum', 'mean']
}).round(2)
revenue_by_star.columns = ['Total_Revenue', 'Avg_Revenue', 'Bookings',
                           'Total_Profit', 'Avg_Profit']

print("\n Revenue by Star Rating:")
print(revenue_by_star)
```

```
ANALYSIS 2: REVENUE PERFORMANCE

 Revenue Performance by Booking Channel:
                 Total_Revenue  Avg_Revenue  Bookings  Total_Profit  \
booking_channel
Web                4.494556e+08     29961.71     15001           0.0
Mobile App         3.541459e+08     29490.04     12009           0.0
Travel Agent       8.863582e+07     29644.09      2990           0.0

                 Avg_Profit  Cancel_Rate
booking_channel
Web                     NaN         15.0
Mobile App              NaN         22.0
Travel Agent            NaN         26.0

 Top 10 Cities by Revenue:
               Total_Revenue  Avg_Revenue  Bookings  Total_Profit  Avg_Profit
city
Chicago          91523337.40     29919.36      3059           0.0         NaN
Los Angeles      91222898.00     30007.53      3040           0.0         NaN
San Francisco    90549893.20     29864.74      3032           0.0         NaN
Las Vegas        90035550.55     29665.75      3035           0.0         NaN
Orlando          89625582.21     29726.56      3015           0.0         NaN
New York         89136957.30     29781.81      2993           0.0         NaN
Boston           88841285.70     29692.94      2992           0.0         NaN
Miami            87958174.93     29725.64      2959           0.0         NaN
Seattle          87091936.60     29623.11      2940           0.0         NaN
Dallas           86251709.50     29387.29      2935           0.0         NaN

 Revenue by Star Rating:
              Total_Revenue  Avg_Revenue  Bookings  Total_Profit  Avg_Profit
star_rating
2              8.873150e+07     29626.55      2995           0.0         NaN
3              3.110431e+08     29736.43     10460           0.0         NaN
4              3.591662e+08     29845.96     12034           0.0         NaN
```

```
5              1.332965e+08      29549.22       4511            0.0          NaN
```

[44]:
```
# STEP 8: ANALYSIS 3 - BOOKING CHANNEL PERFORMANCE

print("ANALYSIS 3: BOOKING CHANNEL DEEP DIVE")

channel_performance = df.groupby('booking_channel').agg({
    'customer_id': 'count',
    'booking_value': ['sum', 'mean'],
    'selling_price': ['sum', 'mean'],
    'profit_margin': ['sum', 'mean'],
    'is_cancelled': 'mean',
    'is_refunded': 'mean',
    'has_coupon': 'mean'
}).round(2)

channel_performance.columns = ['Total_Bookings', 'Total_Booking_Value',␣
 ↪'Avg_Booking_Value',
                               'Total_Revenue', 'Avg_Revenue', 'Total_Profit',␣
 ↪'Avg_Profit',
                               'Cancel_Rate', 'Refund_Rate', 'Coupon_Usage']

channel_performance['Cancel_Rate'] = channel_performance['Cancel_Rate'] * 100
channel_performance['Refund_Rate'] = channel_performance['Refund_Rate'] * 100
channel_performance['Coupon_Usage'] = channel_performance['Coupon_Usage'] * 100

print("\n Comprehensive Channel Performance:")
print(channel_performance)
```

```
ANALYSIS 3: BOOKING CHANNEL DEEP DIVE

 Comprehensive Channel Performance:
                Total_Bookings  Total_Booking_Value  Avg_Booking_Value  \
booking_channel
Mobile App               12009         2.564076e+08           21351.29
Travel Agent              2990         7.311738e+07           24453.97
Web                      15001         4.228908e+08           28190.84

                Total_Revenue  Avg_Revenue  Total_Profit  Avg_Profit  \
booking_channel
Mobile App       3.541459e+08     29490.04           0.0         NaN
Travel Agent     8.863582e+07     29644.09           0.0         NaN
Web              4.494556e+08     29961.71           0.0         NaN

                Cancel_Rate  Refund_Rate  Coupon_Usage
booking_channel
Mobile App             22.0         78.0          21.0
Travel Agent           26.0         77.0          20.0
```

```
Web                    15.0          77.0          21.0
```

[45]:
```python
# STEP 9: ANALYSIS 4 - TEMPORAL TRENDS

print("ANALYSIS 4: TEMPORAL TRENDS")
# Monthly booking trends
df['month_year'] = df['booking_date'].dt.to_period('M')
monthly_trends = df.groupby('month_year').agg({
    'customer_id': 'count',
    'booking_value': 'sum',
    'selling_price': 'sum',
    'is_cancelled': 'mean'
}).round(2)
monthly_trends.columns = ['Total_Bookings', 'Total_Booking_Value',
                          'Total_Revenue', 'Cancel_Rate']
monthly_trends['Cancel_Rate'] = monthly_trends['Cancel_Rate'] * 100

print("\n Monthly Booking Trends:")
print(monthly_trends.head(12))

# Day of week analysis
df['booking_day'] = df['booking_date'].dt.day_name()
day_analysis = df.groupby('booking_day').agg({
    'customer_id': 'count',
    'selling_price': 'mean',
    'is_cancelled': 'mean'
}).round(2)
day_analysis.columns = ['Total_Bookings', 'Avg_Revenue', 'Cancel_Rate']
day_analysis['Cancel_Rate'] = day_analysis['Cancel_Rate'] * 100

print("\n Booking Patterns by Day of Week:")
print(day_analysis)
```

```
ANALYSIS 4: TEMPORAL TRENDS

 Monthly Booking Trends:
            Total_Bookings  Total_Booking_Value  Total_Revenue  Cancel_Rate
month_year
2024-01                686          16445017.04    20397816.50         18.0
2024-02                684          16653266.73    20796930.85         18.0
2024-03                685          16668630.47    20653756.45         22.0
2024-04                686          16596805.95    20247364.00         20.0
2024-05                685          16431309.54    20370587.80         19.0
2024-06                685          16537763.64    20353419.20         18.0
2024-07                686          16846782.76    20510781.10         21.0
2024-08                685          16677784.08    20305897.10         21.0
2024-09                685          16401606.26    20064882.31         20.0
2024-10                685          16474114.42    20340487.15         15.0
```

```
2024-11                686          16278935.15    20389631.35         19.0
2024-12                685          16678245.91    20677608.38         23.0

 Booking Patterns by Day of Week:
             Total_Bookings  Avg_Revenue  Cancel_Rate
booking_day
Friday                1674     29737.16         18.0
Monday                1754     29403.87         19.0
Saturday              1598     30017.73         19.0
Sunday                1599     29745.57         20.0
Thursday              1830     29537.53         18.0
Tuesday               1748     29922.15         19.0
Wednesday             1676     29901.10         20.0
```

[46]:
```
# STEP 10: ANALYSIS 5 - STAY TYPE COMPARISON

print("ANALYSIS 5: LEISURE VS BUSINESS TRAVEL")

stay_comparison = df.groupby('stay_type').agg({
    'customer_id': 'count',
    'booking_value': ['sum', 'mean'],
    'selling_price': ['sum', 'mean'],
    'actual_length_stay': 'mean',
    'is_cancelled': 'mean',
    'is_weekend_checkin': 'mean'
}).round(2)

stay_comparison.columns = ['Total_Bookings', 'Total_Booking_Value',␣
 ↪'Avg_Booking_Value',
                           'Total_Revenue', 'Avg_Revenue', 'Avg_Stay_Length',
                           'Cancel_Rate', 'Weekend_Checkin_Pct']

stay_comparison['Cancel_Rate'] = stay_comparison['Cancel_Rate'] * 100
stay_comparison['Weekend_Checkin_Pct'] = stay_comparison['Weekend_Checkin_Pct']␣
 ↪* 100

print("\n Leisure vs Business Travel Comparison:")
print(stay_comparison)
```

```
ANALYSIS 5: LEISURE VS BUSINESS TRAVEL

 Leisure vs Business Travel Comparison:
            Total_Bookings  Total_Booking_Value  Avg_Booking_Value  \
stay_type
Business             11890         2.989672e+08           25144.42
Leisure              18110         4.534486e+08           25038.58

            Total_Revenue  Avg_Revenue  Avg_Stay_Length  Cancel_Rate  \
```

```
stay_type
Business    3.559278e+08     29935.06             4.03         17.0
Leisure     5.363095e+08     29614.00             3.99         20.0

           Weekend_Checkin_Pct
stay_type
Business                  35.0
Leisure                   35.0
```

[47]:
```python
# STEP 11: ANALYSIS 6 - PAYMENT METHOD ANALYSIS

print("ANALYSIS 6: PAYMENT METHOD PREFERENCES")

payment_analysis = df.groupby('payment_method').agg({
    'customer_id': 'count',
    'selling_price': ['sum', 'mean'],
    'is_cancelled': 'mean',
    'is_refunded': 'mean'
}).round(2)

payment_analysis.columns = ['Total_Bookings', 'Total_Revenue', 'Avg_Revenue',
                            'Cancel_Rate', 'Refund_Rate']
payment_analysis['Cancel_Rate'] = payment_analysis['Cancel_Rate'] * 100
payment_analysis['Refund_Rate'] = payment_analysis['Refund_Rate'] * 100
payment_analysis = payment_analysis.sort_values('Total_Bookings',␣
 ↪ascending=False)

print("\n Payment Method Analysis:")
print(payment_analysis)
```

```
ANALYSIS 6: PAYMENT METHOD PREFERENCES

 Payment Method Analysis:
                Total_Bookings  Total_Revenue  Avg_Revenue  Cancel_Rate  \
payment_method
Debit Card                7603   2.264065e+08     29778.57         19.0
PayPal                    7584   2.251500e+08     29687.51         20.0
Credit Card               7492   2.224251e+08     29688.35         20.0
Bank Transfer             7321   2.182557e+08     29812.28         18.0

                Refund_Rate
payment_method
Debit Card             76.0
PayPal                 78.0
Credit Card            78.0
Bank Transfer          78.0
```

[48]:
```
# STEP 12: ADVANCED INSIGHTS

print("ADVANCED INSIGHTS")
# Insight 1: Lead Time Analysis
df['lead_time'] = (df['check_in_date'] - df['booking_date']).dt.days
lead_time_segments = pd.cut(df['lead_time'], bins=[-1, 7, 30, 90, 365],
                            labels=['Last Minute (0-7 days)', 'Short (8-30␣
 ↪days)',
                                    'Medium (31-90 days)', 'Long (90+ days)'])
df['lead_time_segment'] = lead_time_segments

lead_time_analysis = df.groupby('lead_time_segment').agg({
    'customer_id': 'count',
    'selling_price': 'mean',
    'is_cancelled': 'mean'
}).round(2)
lead_time_analysis.columns = ['Bookings', 'Avg_Revenue', 'Cancel_Rate']
lead_time_analysis['Cancel_Rate'] = lead_time_analysis['Cancel_Rate'] * 100

print("\n Lead Time vs Cancellation Analysis:")
print(lead_time_analysis)

# Insight 2: High Value Customer Identification
high_value_threshold = df['selling_price'].quantile(0.75)
df['is_high_value'] = df['selling_price'] >= high_value_threshold

high_value_profile = df.groupby('is_high_value').agg({
    'customer_id': 'count',
    'selling_price': 'mean',
    'actual_length_stay': 'mean',
    'is_cancelled': 'mean',
    'star_rating': 'mean'
}).round(2)
high_value_profile.columns = ['Count', 'Avg_Revenue', 'Avg_Stay',␣
 ↪'Cancel_Rate', 'Avg_Star_Rating']
high_value_profile['Cancel_Rate'] = high_value_profile['Cancel_Rate'] * 100

print("\n High Value Customer Profile:")
print(high_value_profile)

# Insight 3: Room Type Profitability
room_profitability = df.groupby('room_type').agg({
    'customer_id': 'count',
    'profit_margin': ['sum', 'mean'],
    'profit_margin_pct': 'mean',
    'is_cancelled': 'mean'
}).round(2)
```

```
room_profitability.columns = ['Bookings', 'Total_Profit', 'Avg_Profit',
                              'Profit_Margin_Pct', 'Cancel_Rate']
room_profitability['Cancel_Rate'] = room_profitability['Cancel_Rate'] * 100

print("\n Room Type Profitability:")
print(room_profitability)
```

```
ADVANCED INSIGHTS

 Lead Time vs Cancellation Analysis:
                        Bookings  Avg_Revenue  Cancel_Rate
lead_time_segment
Last Minute (0-7 days)       213     29084.26          5.0
Short (8-30 days)            624     29270.08          4.0
Medium (31-90 days)         1538     29668.76          4.0
Long (90+ days)             3141     29922.74          4.0

 High Value Customer Profile:
                Count  Avg_Revenue  Avg_Stay  Cancel_Rate  Avg_Star_Rating
is_high_value
False           22500     24548.01      4.01         19.0             3.60
True             7500     45320.94      4.00         19.0             3.61

 Room Type Profitability:
            Bookings  Total_Profit  Avg_Profit  Profit_Margin_Pct  Cancel_Rate
room_type
Deluxe         10478           0.0         NaN               31.0         16.0
Standard       16552           0.0         NaN               31.0         22.0
Suite           2970           0.0         NaN               31.0         16.0
```

[49]:
```
# STEP 13: EXPORT RESULTS TO CSV

print("EXPORTING ANALYSIS RESULTS")

# Create exports directory if not exists
import os
if not os.path.exists('analysis_outputs'):
    os.makedirs('analysis_outputs')

# Export all analysis results
cancel_by_channel.to_csv('analysis_outputs/cancellation_by_channel.csv')
revenue_by_channel.to_csv('analysis_outputs/revenue_by_channel.csv')
channel_performance.to_csv('analysis_outputs/channel_performance.csv')
monthly_trends.to_csv('analysis_outputs/monthly_trends.csv')
stay_comparison.to_csv('analysis_outputs/stay_type_comparison.csv')
payment_analysis.to_csv('analysis_outputs/payment_analysis.csv')
room_profitability.to_csv('analysis_outputs/room_profitability.csv')
```

```
print(" Analysis results exported to 'analysis_outputs' folder")
```

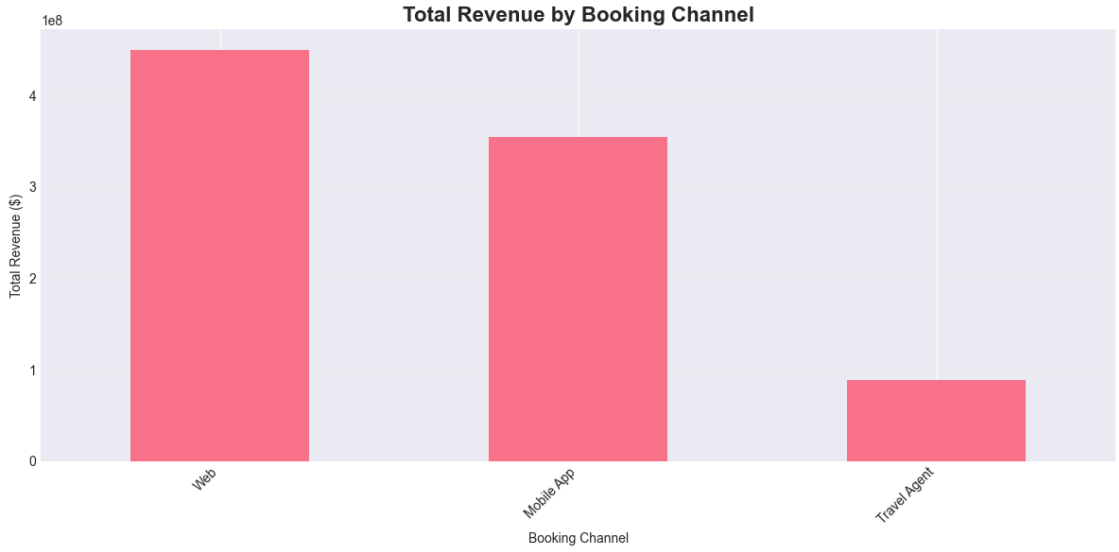
```
EXPORTING ANALYSIS RESULTS
 Analysis results exported to 'analysis_outputs' folder
```

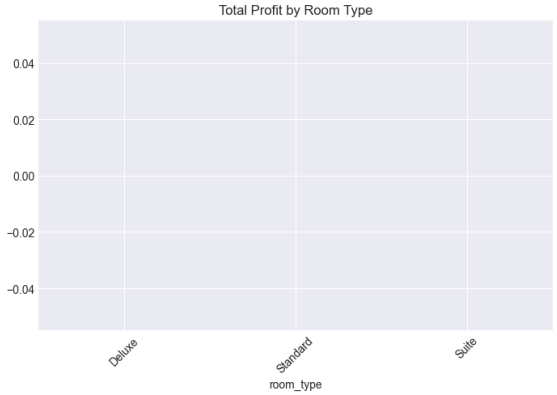
[50]:
```
# STEP 14: CREATE VISUALIZATIONS
import matplotlib.pyplot as plt

print("GENERATING VISUALIZATIONS")

%matplotlib inline

plt.rcParams['figure.figsize'] = (12, 8)
plt.rcParams['font.size'] = 10

# ---------- 1. Cancellation Rate by Channel ----------
plt.figure(figsize=(12, 6))
cancel_by_channel['Cancellation_Rate'].plot(kind='bar')
plt.title('Cancellation Rate by Booking Channel', fontsize=16,␣
 ↪fontweight='bold')
plt.xlabel('Booking Channel')
plt.ylabel('Cancellation Rate (%)')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', alpha=0.3)
plt.tight_layout()
plt.show()
plt.savefig('analysis_outputs/viz_cancellation_by_channel.png', dpi=300)
plt.close()

# ---------- 2. Revenue by Channel ----------
plt.figure(figsize=(12, 6))
revenue_by_channel['Total_Revenue'].plot(kind='bar')
plt.title('Total Revenue by Booking Channel', fontsize=16, fontweight='bold')
plt.xlabel('Booking Channel')
plt.ylabel('Total Revenue ($)')
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', alpha=0.3)
plt.tight_layout()
plt.show()
plt.savefig('analysis_outputs/viz_revenue_by_channel.png', dpi=300)
plt.close()

# ---------- 3. Monthly Trends ----------
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
```

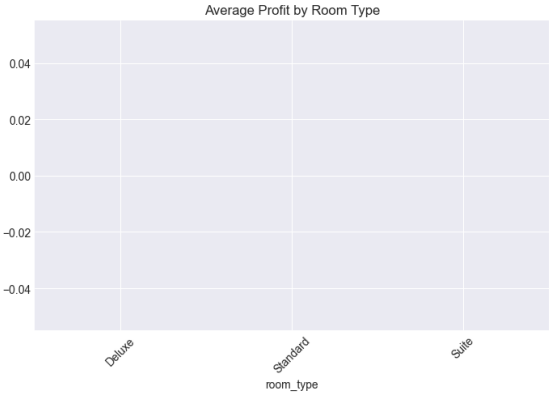
```
monthly_trends['Total_Bookings'].plot(marker='o')
plt.title('Monthly Booking Trends')
plt.grid(alpha=0.3)

plt.subplot(1, 2, 2)
monthly_trends['Cancel_Rate'].plot(marker='o')
plt.title('Monthly Cancellation Rate')
plt.grid(alpha=0.3)

plt.tight_layout()
plt.show()
plt.savefig('analysis_outputs/viz_monthly_trends.png', dpi=300)
plt.close()

# ---------- 4. Room Type Performance ----------
fig, axes = plt.subplots(2, 2, figsize=(14, 10))

room_profitability['Total_Profit'].plot(kind='bar', ax=axes[0, 0])
axes[0, 0].set_title('Total Profit by Room Type')

room_profitability['Avg_Profit'].plot(kind='bar', ax=axes[0, 1])
axes[0, 1].set_title('Average Profit by Room Type')

room_profitability['Cancel_Rate'].plot(kind='bar', ax=axes[1, 0])
axes[1, 0].set_title('Cancellation Rate by Room Type')

room_profitability['Bookings'].plot(kind='bar', ax=axes[1, 1])
axes[1, 1].set_title('Total Bookings by Room Type')

for ax in axes.flatten():
    ax.tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()
plt.savefig('analysis_outputs/viz_room_type_analysis.png', dpi=300)
plt.close()

print("ANALYSIS COMPLETE!")
```

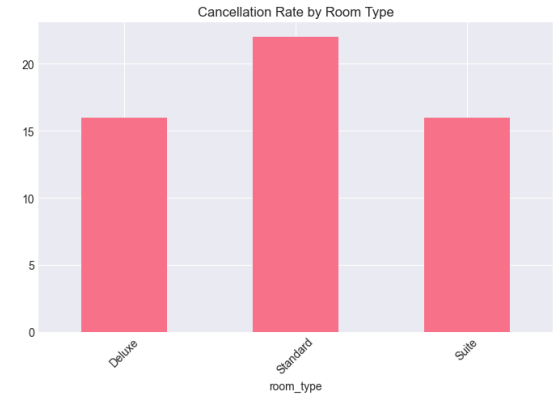
```
GENERATING VISUALIZATIONS
```

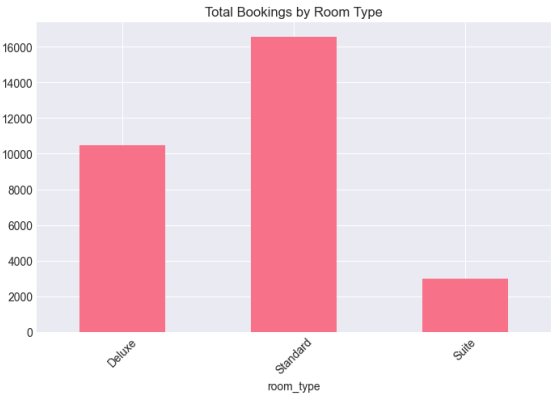

Cancellation Rate by Booking Channel
Cancellation Rate (%)
0
5
10
15
20
25
Travel Agent
Mobile App
Web
Booking Channel


Total Revenue by Booking Channel
1e8
Total Revenue ($)
0
1
2
3
4
Web
Mobile App
Travel Agent
Booking Channel









ANALYSIS COMPLETE!

[ ]: