810Report

2025-05-19

.

Intro

Entrepôt, the 10th arrondissement of Paris, is a vibrant and culturally diverse district with a population of 81,926 (2022). Known historically for its warehouses ("entrepôts") that facilitated trade along the Canal Saint-Martin, the area today is a dynamic hub of multicultural influences, including a large Turkish community ("La Petite Turquie"), as well as Indian, Pakistani, North African, and Caribbean populations. This fusion is reflected in its lively streets, diverse culinary scene—such as Passage Brady's "Little India"—and its appeal to young professionals and students seeking affordability in Paris.

Strategically located, Entrepôt is home to two of Paris' busiest railway stations, Gare du Nord and Gare de l'Est, making it a key transit hub with high foot traffic. The presence of the Canal Saint-Martin further enhances its charm, offering scenic waterways that attract both locals and tourists.

For potential investors considering Airbnb properties in Entrepôt, these are the questions we've decided to look into:

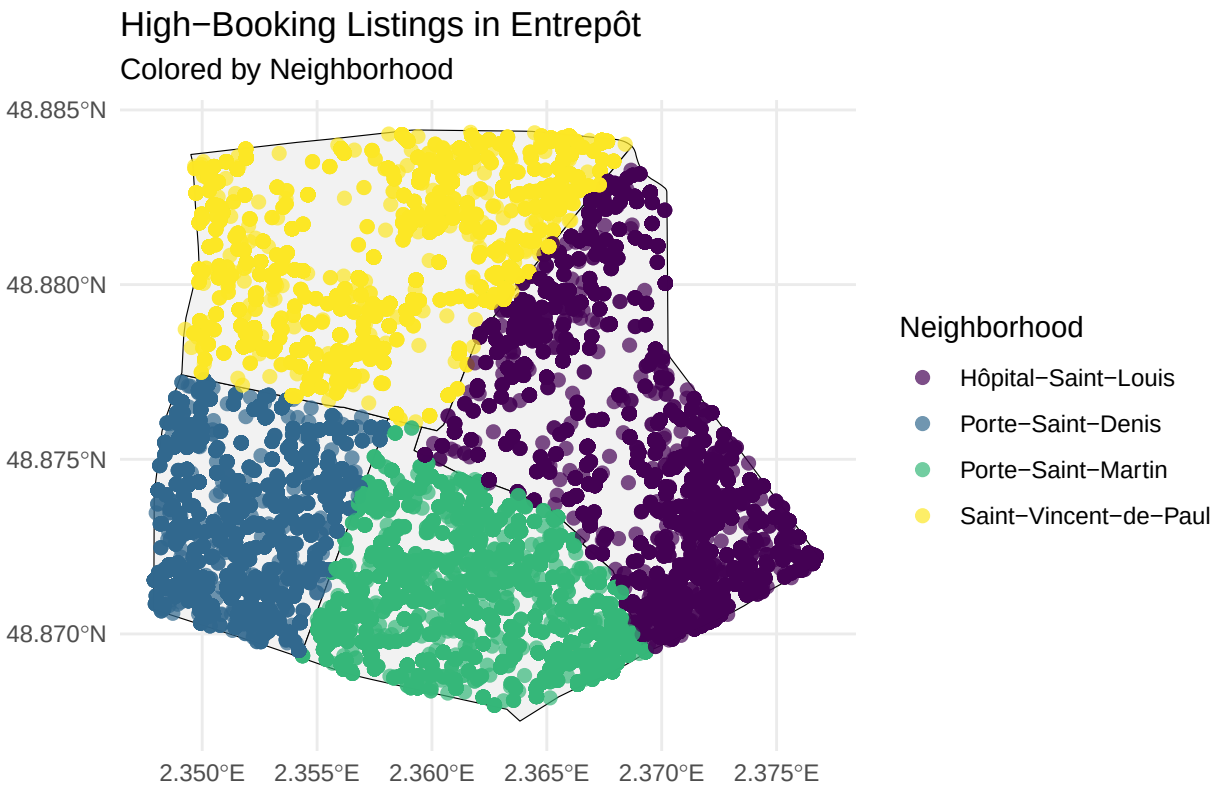
If you're looking to buy property in Entrepôt, which physical characteristics are most important to consider to achieve high booking?

Does proximity to major landmarks (Gare du Nord, Gare de l'Est, Canal Saint-Martin) correlate with higher demand?

Once you've bought property in Entrepôt, what characteristics should you include in your listing to boost it to high booking?

Background Info - Looking Into the Neighborhoods of Entrepôt

```
ggplot() +
  geom_sf(data = quartiers_10, fill = "gray95", color = "black") +
  geom_sf(data = df_labeled, aes(color = l_qu), size = 2, alpha = 0.7) +
  scale_color_viridis_d(name = "Neighborhood") +
  labs(
    title = "High-Booking Listings in Entrepôt",
    subtitle = "Colored by Neighborhood",
  ) +
  theme_minimal()
```




```
print(summary_stats)
```

```
## # A tibble: 4 x 8
##   l_qu     avg_price pct_superhost avg_accommodates listing_count total_listings
##   <chr>        <dbl>         <dbl>            <dbl>         <int>          <int>
## 1 Hôpital~      78.6          4.71             2.81          9024          20999
## 2 Porte-S~      91.2          5.54             2.84          6085          15221
## 3 Porte-S~      90.5          7.44             2.77          7273          20328
## 4 Saint-V~      91.9          3.88             3.02          6757          15648
## # i 2 more variables: high_booking_count <int>, high_booking_pct <dbl>
```

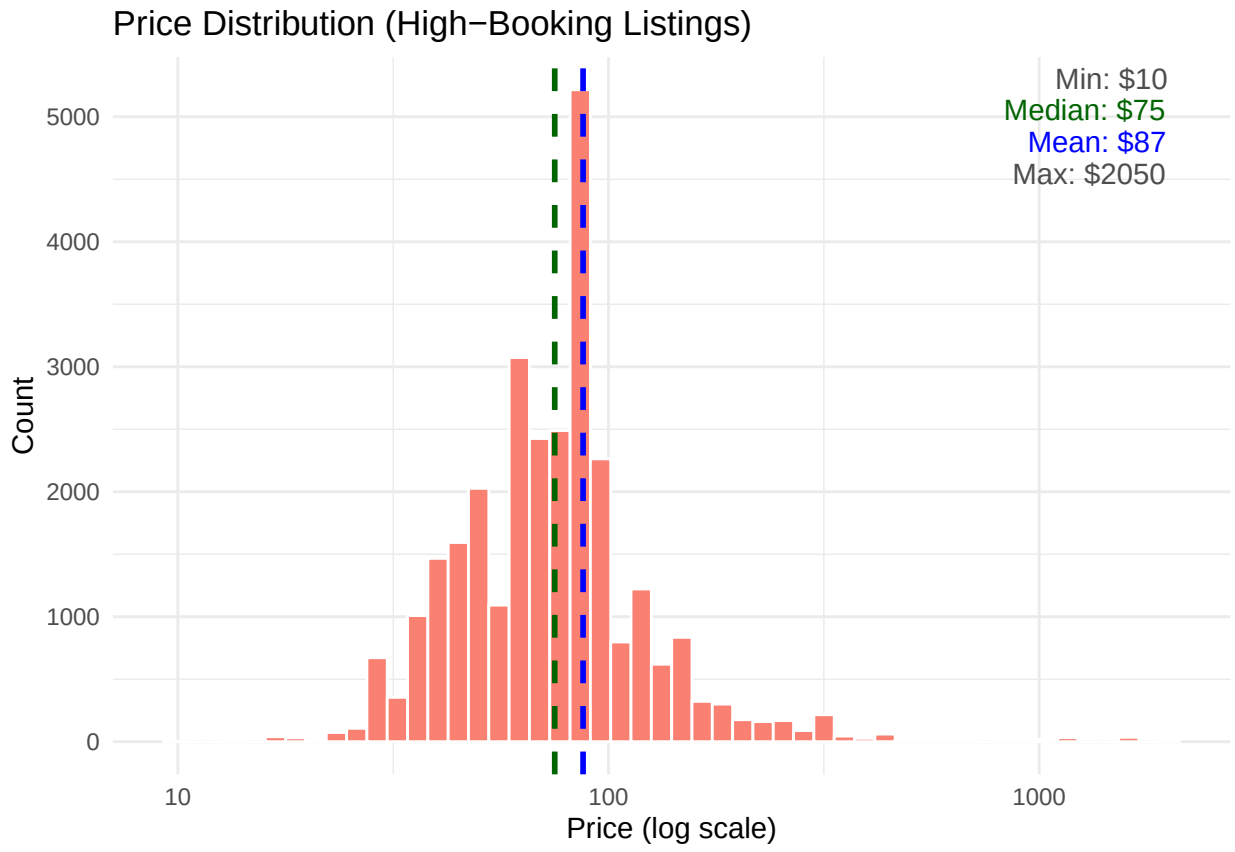
Conducting EDA

The histogram below shows the optimal range by showing where most high-booking properties cluster, which is around the $75 median price. This helps tell us what price point buyers and renters find most attractive. The log scale helps demonstrate that extreme outliers book far less frequently and helps make it so the histogram is not heavily skewed to the right.

```
# Plot Price Distribution
df %>%
  filter(high_booking == 1, price > 0) %>%
  ggplot(aes(price)) +
  geom_histogram(bins = 50, fill = "salmon", color = "white") +
  scale_x_log10() +
```

```
  geom_vline(xintercept = mean_val, color = "blue", linetype = "dashed", size = 1) +
  geom_vline(xintercept = median_val, color = "darkgreen", linetype = "dashed", size = 1) +
  annotate("text", x = max_val, y = Inf, hjust = 1.05, vjust = 1.5,
           label = paste0("Min: $", round(min_val, 0)), color = "gray30", size = 4) +
  annotate("text", x = max_val, y = Inf, hjust = 1.05, vjust = 3,
           label = paste0("Median: $", round(median_val, 0)), color = "darkgreen", size = 4) +
  annotate("text", x = max_val, y = Inf, hjust = 1.05, vjust = 4.5,
           label = paste0("Mean: $", round(mean_val, 0)), color = "blue", size = 4) +
  annotate("text", x = max_val, y = Inf, hjust = 1.05, vjust = 6,
           label = paste0("Max: $", round(max_val, 0)), color = "gray30", size = 4) +
  labs(
    title = "Price Distribution (High-Booking Listings)",
    x = "Price (log scale)",
    y = "Count"
  ) +
  theme_minimal()
```

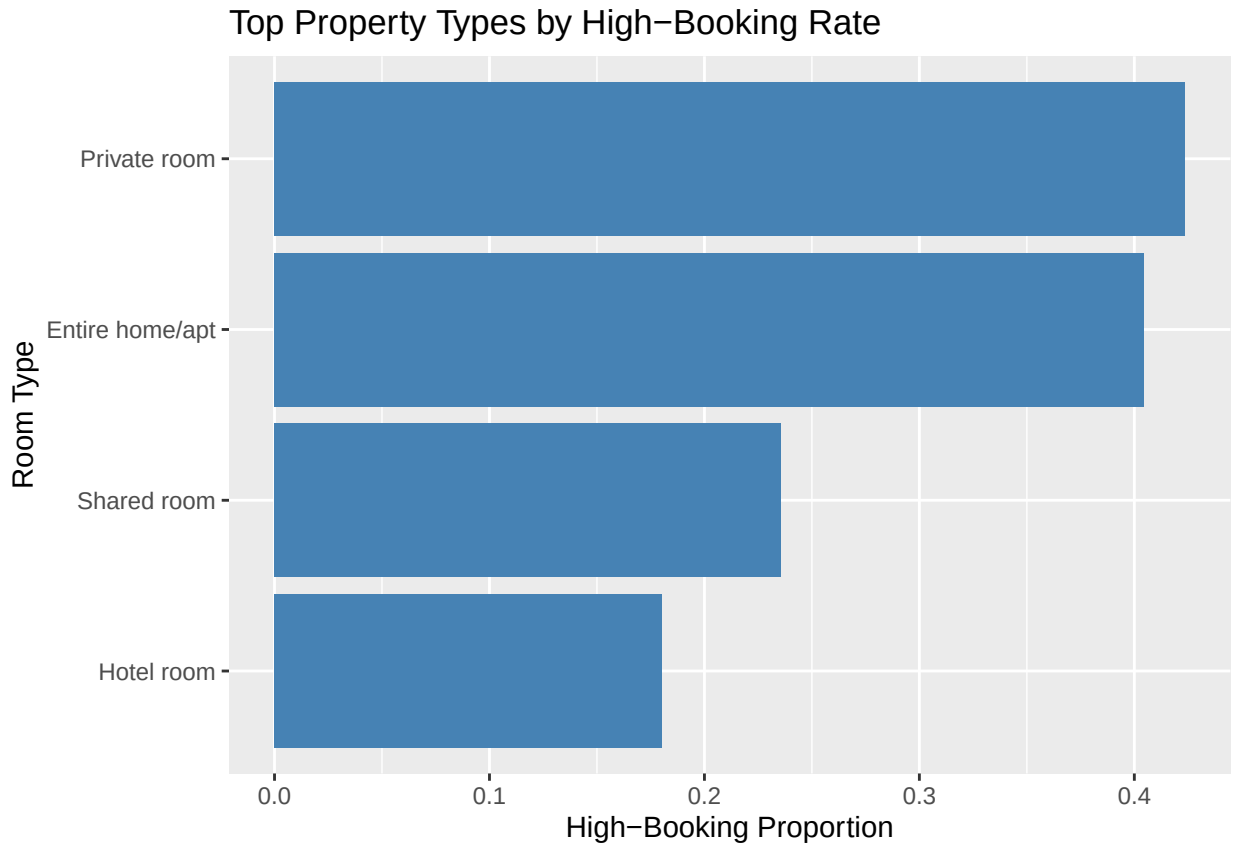
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Next we created a bar chart which reveals that private rooms and entire home/apartments deliver the highest booking ratios, suggesting that privacy and autonomy are what guests prioritize. On the other hand, shared

or hotel rooms underperform, so prioritizing standalone units will likely maximize return when making an investment.

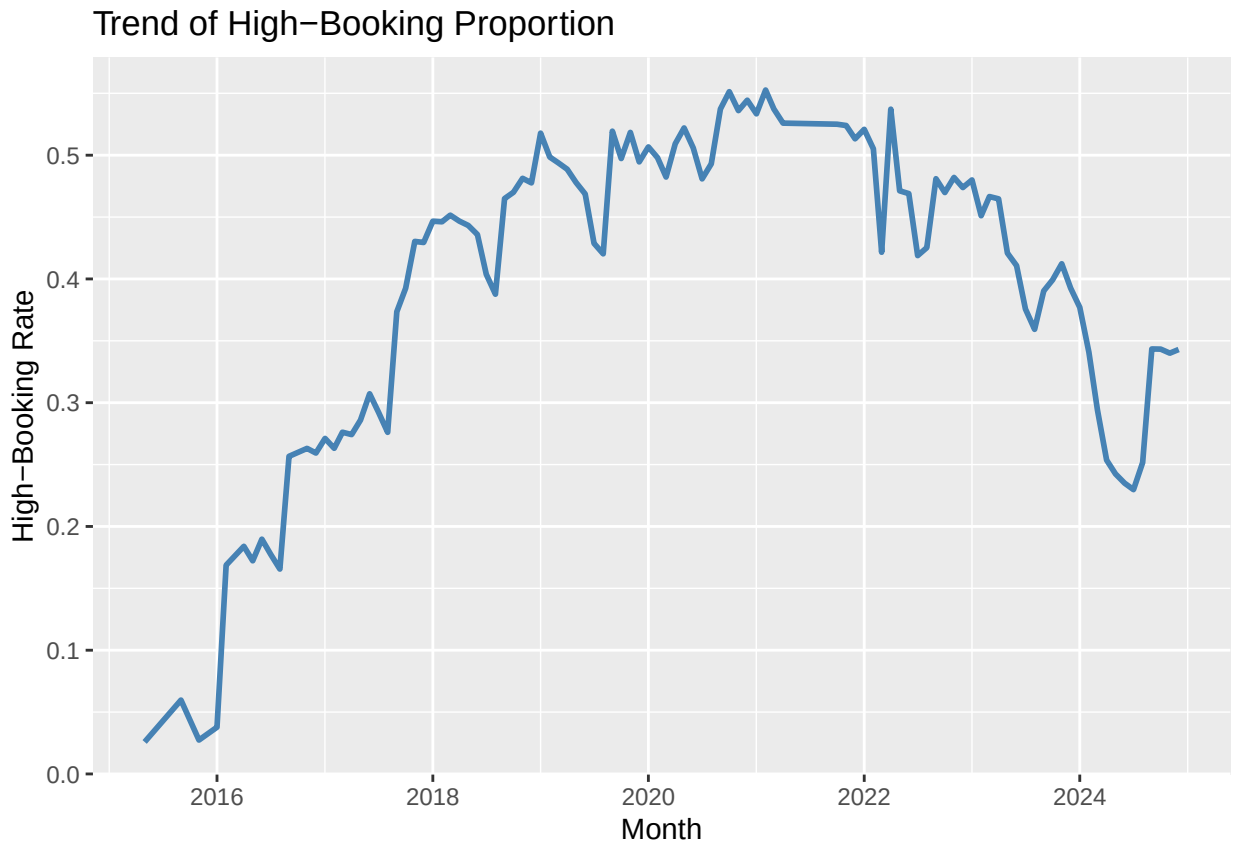
```
df %>%
  group_by(room_type) %>%
  summarise(
    n = n(),
    high_rate = mean(high_booking),
    avg_price = mean(price, na.rm = TRUE)
  ) %>%
  arrange(desc(high_rate)) %>%
  slice(1:10) %>%
  ggplot(aes(
    x = reorder(room_type, high_rate),
    y = high_rate
  )) +
    geom_col(fill = "steelblue") +
    coord_flip() +
    labs(
      title = "Top Property Types by High-Booking Rate",
      x = "Room Type",
      y = "High-Booking Proportion"
    )
```



The line trend chart below shows how booking rates rose steadily through 2020, dipped during early-2024, then began rebounding. This indicates seasonality and broader demand cycles. Timing your market entry

and listing updates around high-demand months can further amplify your property's visibility and booking success.

```
df %>%
  mutate(month = floor_date(ymd(date), "month")) %>%
  group_by(month) %>%
  summarise(high_rate = mean(high_booking)) %>%
  ggplot(aes(x = month, y = high_rate)) +
    geom_line(size = 1, color = "steelblue") +
    labs(x = "Month", y = "High-Booking Rate",
         title = "Trend of High-Booking Proportion")
```



Part of our Exploratory Data Analysis for this project was focused on the variable "host_is_superhost". Our initial assumption was that Superhosts should attract more reservations because guests often filter for superhosts as a signal of reliability, responsiveness, and a high-quality stay. From an investor's standpoint, properties managed by Superhosts would be expected to enjoy above-average occupancy and revenue, so mapping out the relationship between Superhost status and booking performance is a natural first step for EDA. Although, when looking at descriptive statistics related to superhost status, only 19.3% of unique listings in the Entrepôt sample ever achieve Superhost status, and among those that do, not all translate that reputation into consistent "high_booking" performance. In fact, the numbers show that only 7.4% of Superhost listings fall into the highest-booking category, and many top-performing properties are not Superhost at all. By focusing on this variable, we can uncover if other factors such as pricing strategy, seasonal availability, or neighborhood dynamics are stronger drivers of demand than just Superhost status.

```r
# Unique IDs with high_booking == 1
cat("Total unique IDs:", total_ids, "\n",
    "High-booking IDs:", high_ids, "\n",
    "Percent high-booking IDs:", round(pct_high_ids, 2), "%\n")
```

```
## Total unique IDs: 3720
##  High-booking IDs: 1563
##  Percent high-booking IDs: 42.02 %
```

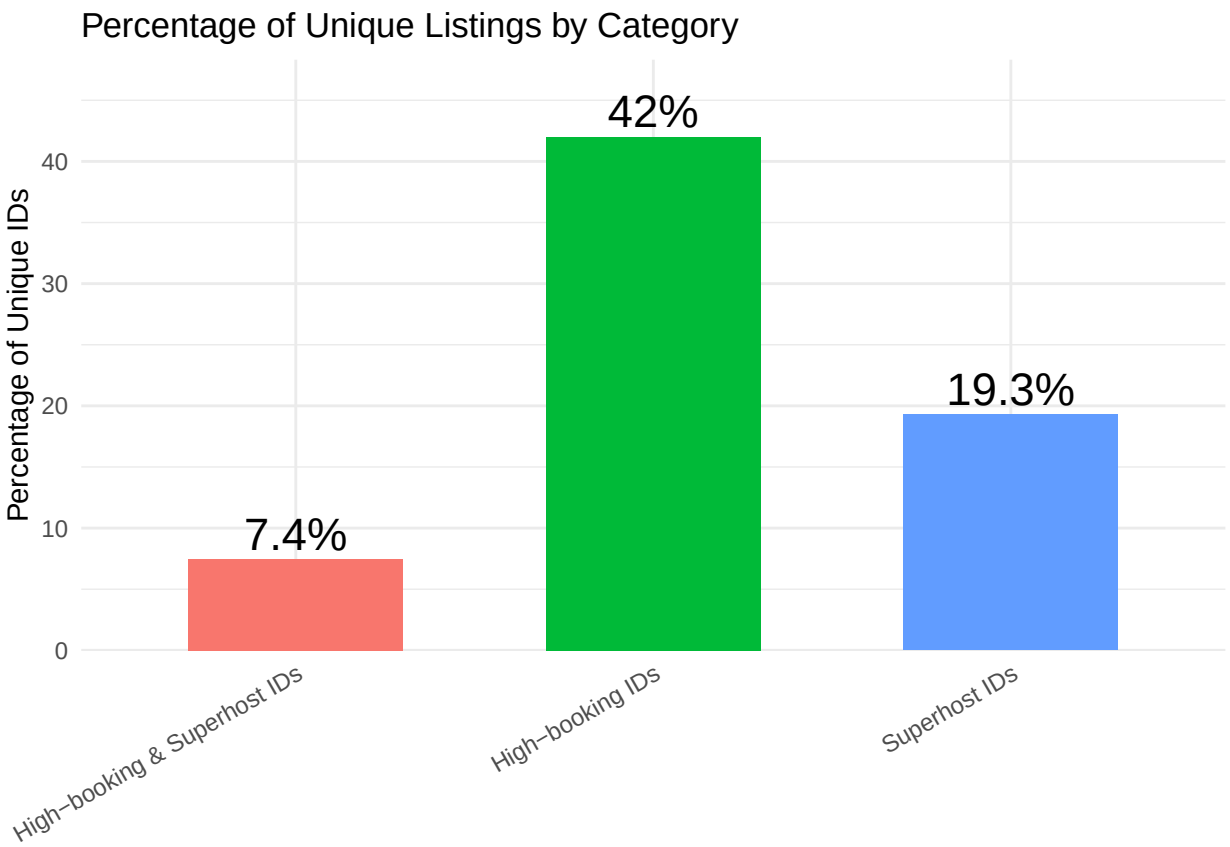
```r
# Unique IDs where host_is_superhost == 1
cat(
  "Total unique IDs:       ", total_ids,     "\n",
  "Superhost unique IDs:  ", superhost_ids, "\n",
  "Percent superhost IDs: ", round(pct_superhosts, 2), "%\n"
)
```

```
## Total unique IDs:        3720
##  Superhost unique IDs:   718
##  Percent superhost IDs:  19.3 %
```

```r
# Unique IDs where both high_booking == 1 AND host_is_superhost == 1
cat(
  "Total unique IDs:                ", total_ids, "\n",
  "High-booking & Superhost IDs:   ", both_ids, "\n",
  "Percent satisfying both:        ", round(pct_both, 2), "%\n"
)
```

```
## Total unique IDs:                 3720
##  High-booking & Superhost IDs:    277
##  Percent satisfying both:         7.45 %
```

```r
# Bar chart
ggplot(stats, aes(x = metric, y = percentage, fill = metric)) +
  geom_col(width = 0.6, show.legend = FALSE) +
  geom_text(aes(label = paste0(round(percentage, 1), "%")),
            vjust = -0.3,
            size  = 6) +
  scale_y_continuous(
    expand = expansion(mult = c(0, 0.15))
  ) +
  labs(
    title = "Percentage of Unique Listings by Category",
    x     = NULL,
    y     = "Percentage of Unique IDs"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 30, hjust = 1)
  )
```



Enough EDA, Let's Create Some DAGs

DAGs clarify how different property features—like location, amenities, and pricing—actually influence Airbnb performance in Entrepôt, separating true drivers of demand from misleading correlations. By mapping these cause-and-effect relationships, DAGs help investors avoid costly oversights (e.g., overvaluing "luxury" features that don't impact bookings) and instead focus on high-impact factors, such as proximity to transit or micro-neighborhood trends. This ensures data-driven decisions maximize occupancy rates and ROI, not just guesswork.

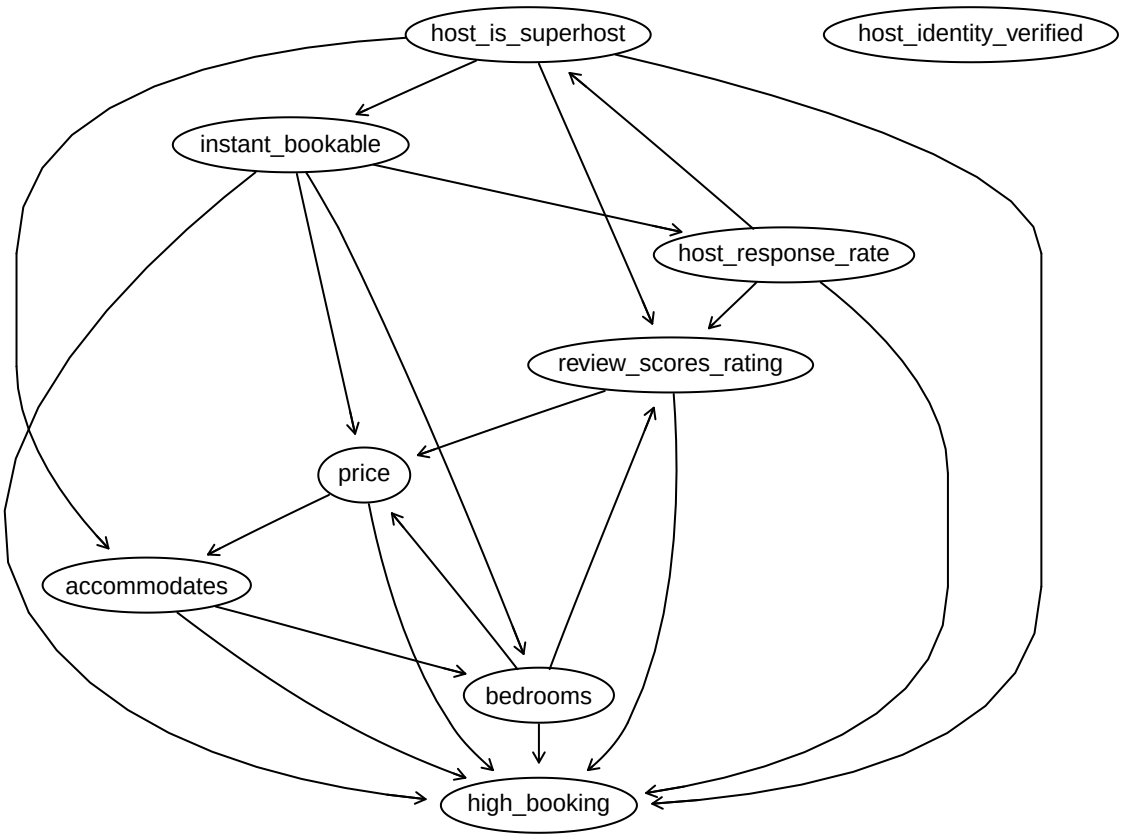
```
# Convert logical fields to binary (1 = TRUE, 0 = FALSE)
data <- df %>%
  mutate(
    host_identity_verified = ifelse(host_identity_verified == "TRUE", 1, 0),
    host_has_profile_pic = ifelse(host_has_profile_pic == "TRUE", 1, 0)
  )

# Convert host_response_rate to numeric if it's a string like "90%"
data$host_response_rate <- as.numeric(gsub("%", "", data$host_response_rate))

# Convert host_response_time to factor with meaningful labels (if needed)
# Optional: skip this if already numeric in your dataset
#data$host_response_time <- as.factor(data$host_response_time)

# Handle missing values: remove rows with NA in key variables (or you can impute)
key_vars3 <- c(
```

```
  "high_booking", "host_is_superhost", "instant_bookable",
  "host_response_rate", "host_identity_verified",
   "accommodates", "bedrooms",
  "review_scores_rating", "price"
)

data_selected = data[, key_vars3]

suffStat <- list(C = cor(data_selected), n = nrow(data_selected))

varNames <- colnames(data_selected)

skel.entrepot <- pcalg::skeleton(suffStat, indepTest = gaussCItest, labels = varNames, alpha = 0.05)

#mygraph(skel.entrepot)

start_time <- Sys.time()

pc.entrepot <- pc(suffStat, indepTest = gaussCItest, labels = varNames, alpha = 0.01)

end_time <- Sys.time()

end_time - start_time
```

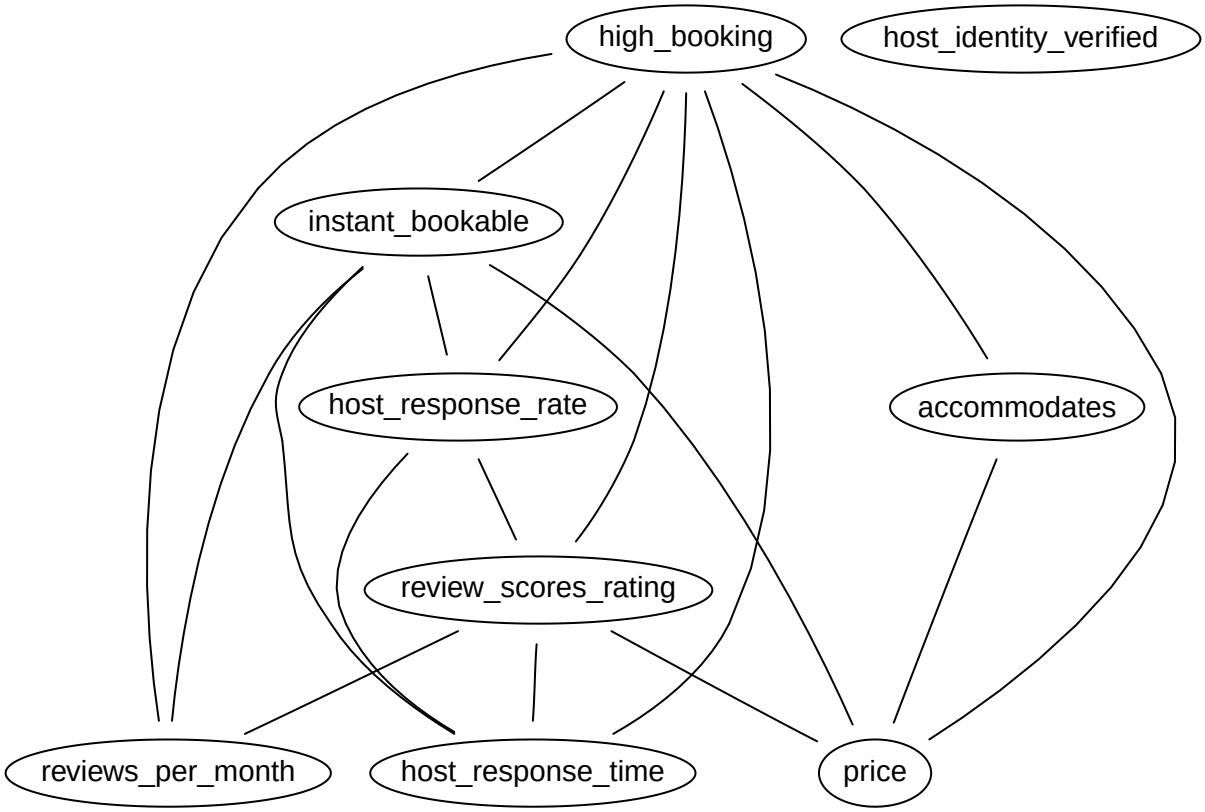
```
## Time difference of 0.06506896 secs
```

Below is an initial DAG created to try and discover any causal relationships with the high_booking variable. From this DAG, we can see that there are 4 potential causes for high_booking: host_is_superhost, accommodates, host_identity_verified and instant_bookable. While we do not know for certain if these variables cause high_booking because of potential hidden colliders or confounders, this DAG can give use a good starting point into discovering causal relationships. Additionally, this DAG can give us insight to understand that price does not have a deterministic relationship with high_booking.

```
mygraph(pc.entrepot)
```




```
key_vars6 <- c(
  "high_booking", "instant_bookable",
  "host_response_rate", "host_identity_verified",
   "accommodates",
  "review_scores_rating", "price", "reviews_per_month",
  "host_response_time"
)

data_selected = data[, key_vars6]

suffStat <- list(C = cor(data_selected), n = nrow(data_selected))

varNames <- colnames(data_selected)

skel.entrepot4 <- pcalg::skeleton(suffStat, indepTest = gaussCItest, labels = varNames, alpha = 0.05)

mygraph(skel.entrepot4)
```

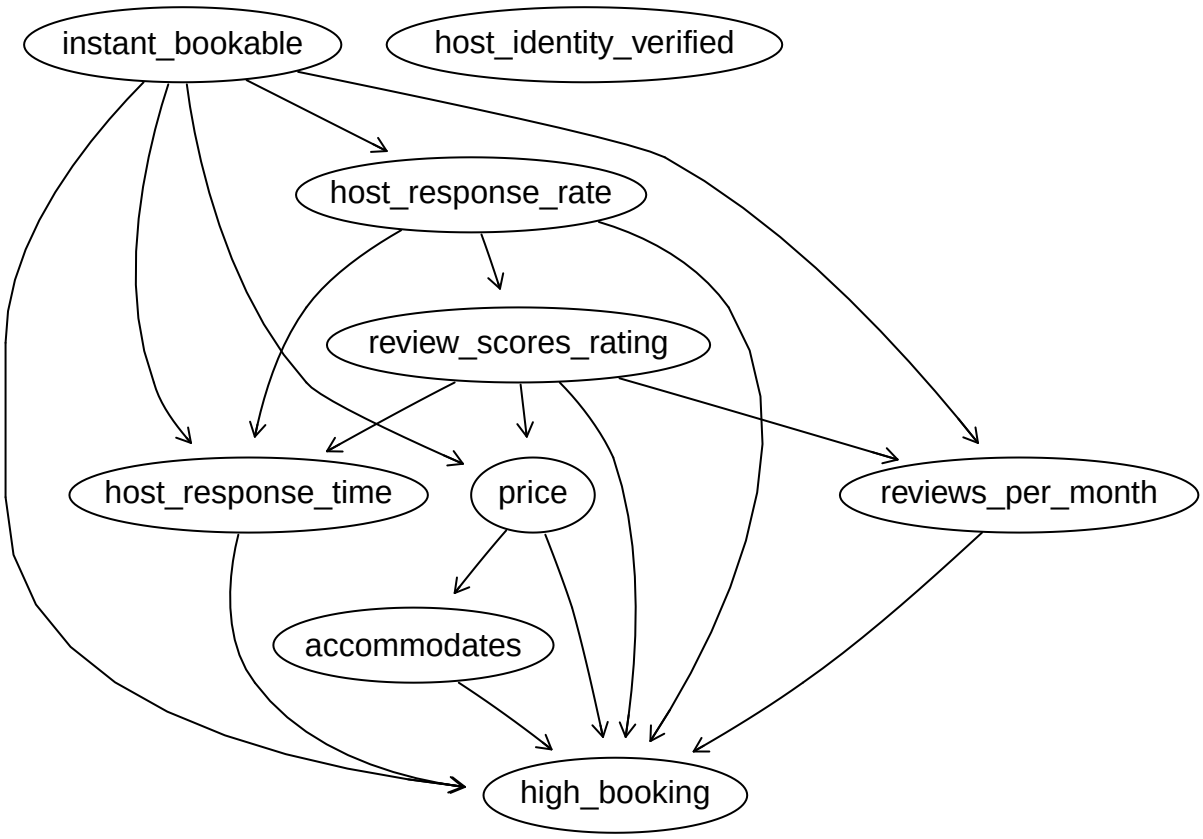



```
start_time <- Sys.time()

pc.entrepot4 <- pc(suffStat, indepTest = gaussCItest, labels = varNames, alpha = 0.01)

end_time <- Sys.time()

end_time - start_time
```

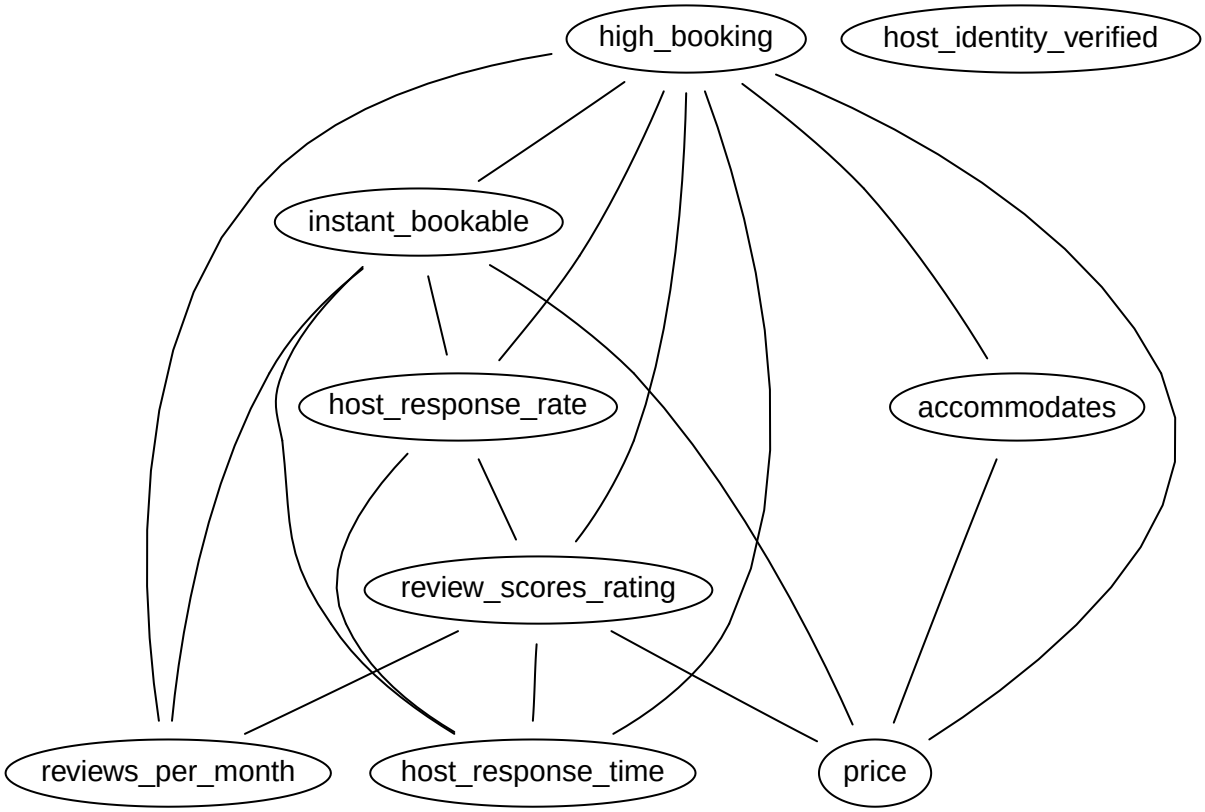
```
## Time difference of 0.04709697 secs
```

This next DAG uses the different room types variables and also removes the superhost status. We chose to remove the superhost status because we deemed it may be irrelevant from earlier analysis. The main takeaway from this diagram is that host_response_rate may have causal influence on high_booking. While we found that a specific roomtype has no causal influence on high_booking, we were ultimately wrong to include these variables in the way we did. The room types variables should have been set as exogenous variables, since so other variable can change their status.

```
mygraph(pc.entrepot4)
```




```
key_vars6 <- c(
  "high_booking", "instant_bookable",
  "host_response_rate", "host_identity_verified",
   "accommodates",
  "review_scores_rating", "price", "reviews_per_month",
  "host_response_time"
)

data_selected = data[, key_vars6]

suffStat <- list(C = cor(data_selected), n = nrow(data_selected))

varNames <- colnames(data_selected)

skel.entrepot4 <- pcalg::skeleton(suffStat, indepTest = gaussCItest, labels = varNames, alpha = 0.05)

mygraph(skel.entrepot4)
```

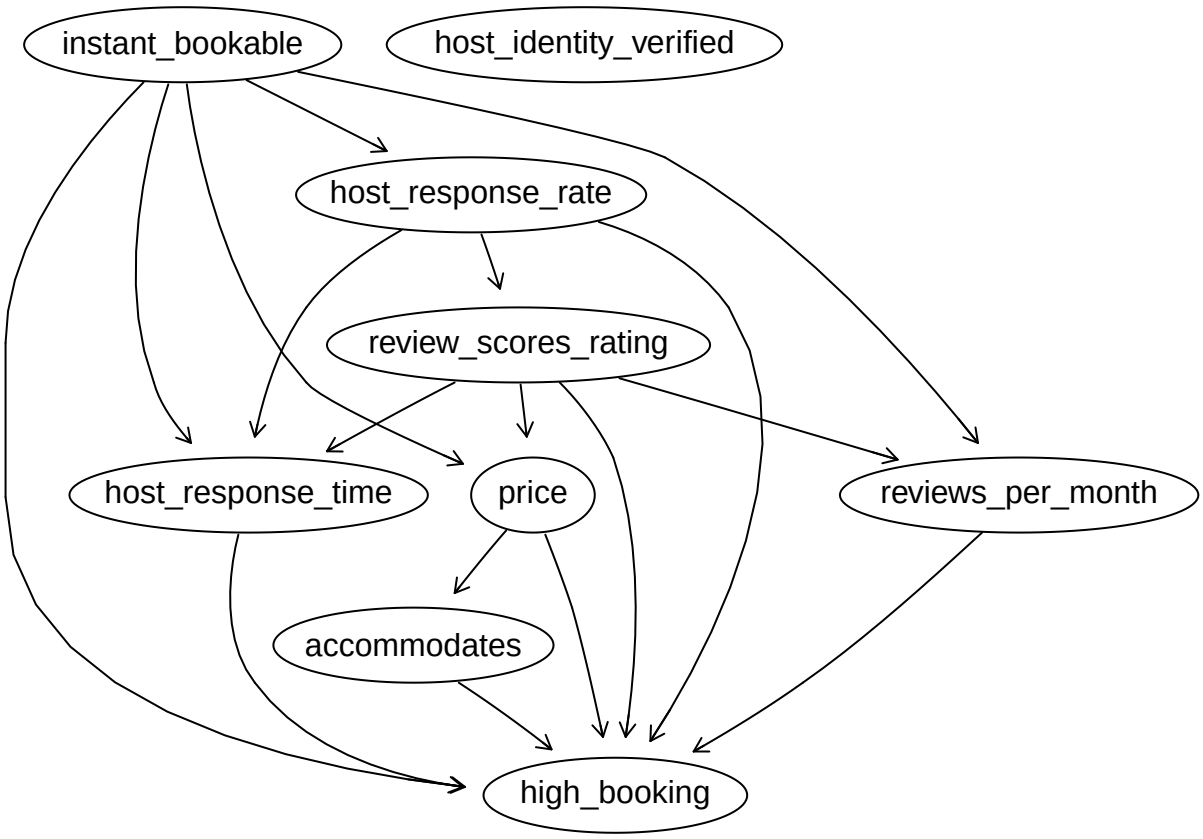



```
start_time <- Sys.time()

pc.entrepot4 <- pc(suffStat, indepTest = gaussCItest, labels = varNames, alpha = 0.01)

end_time <- Sys.time()

end_time - start_time
```

```
## Time difference of 0.04258585 secs
```

This final DAG adds the reviews_per_month and host_response_time variables. The main takeaways from the additions here is that accommodates, instantly_bookable and host_identity seem to be most important in causing high_booking, as these variables have shown up multiple times in the DAGs. We also discovered that reviews_per_month has a potential causal relationship with high_booking. Overall, these DAGs we created were used to gain a visual representation of which variables are most important when considering how we can predict an Airbnb listing is high_booking or not.

```
mygraph(pc.entrepot4)
```



Looking at Conditional Average Treatment Effect of Variables Across Different Price Ranges

For investors evaluating Airbnb properties in Entrepôt, Conditional Average Treatment Effect (CATE) modeling unlocks precision by revealing how the impact of key features varies across market segments—not just on average. Unlike traditional analytics that might suggest "walkability always boosts bookings," CATE quantifies where and for whom it matters most (e.g., a 15% occupancy lift in mid-tier units near Gare du Nord, but just 5% in luxury properties). This allows investors to:

Target upgrades strategically (e.g., prioritize parking in budget areas, not high-end),

Optimize pricing tiers (balconies command premiums in scenic zones but add little value elsewhere), and

Identify undervalued micro-markets (e.g., "workspace" appeals disproportionately to digital nomads in specific neighborhoods).

Below you'll see our CATE analysis using distances to key landmarks as well as super host and accommodates.

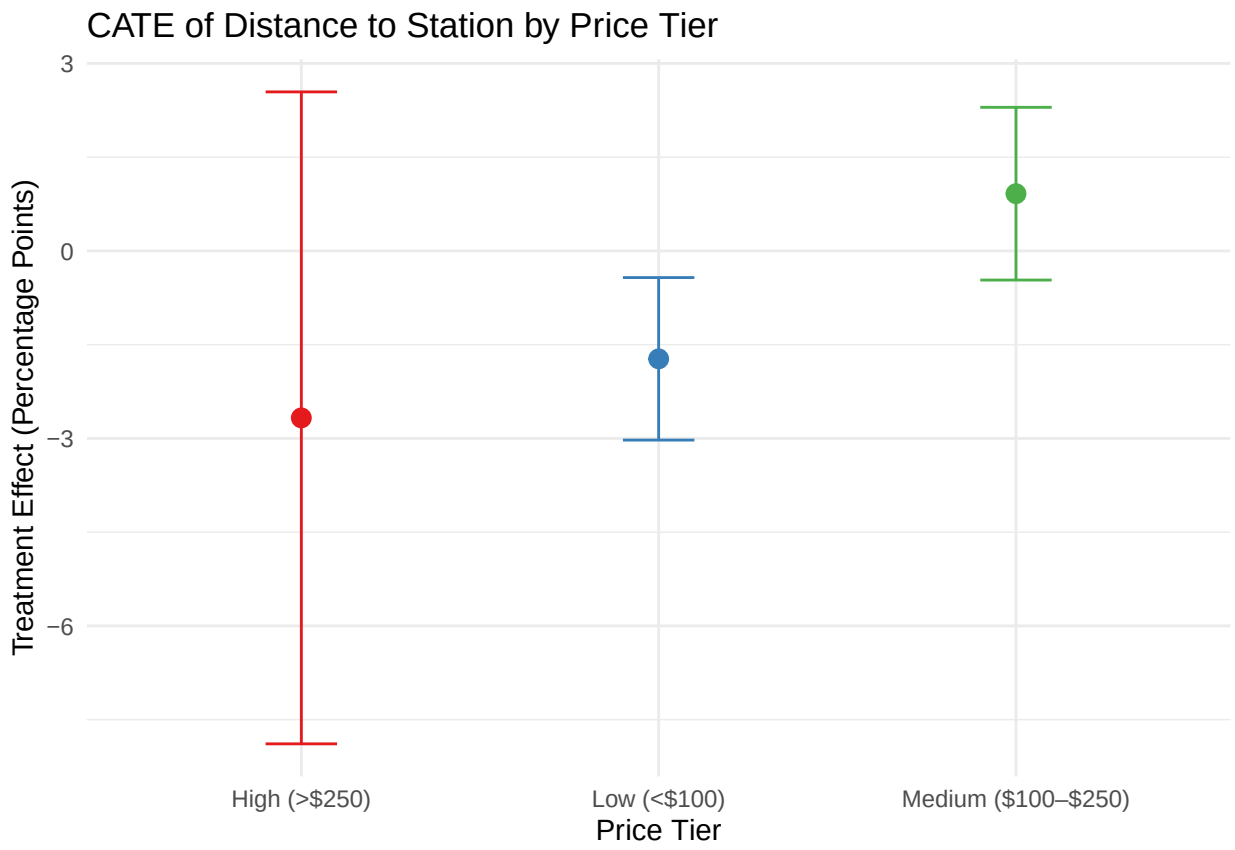
Distance to Train Station:

```
print(paste("CATE (All):", round(ate[1], 4), "| SE:", round(ate[2], 4)))
```

```
## [1] "CATE (All): -0.9593 | SE: 0.5185"
```

```
ggplot(cate_results, aes(x = Price_Tier, y = CATE, color = Price_Tier)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI), width = 0.2) +
  labs(
```

```
    title = "CATE of Distance to Station by Price Tier",
    x = "Price Tier",
    y = "Treatment Effect (Percentage Points)"
  ) +
  theme_minimal() +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "none")
```

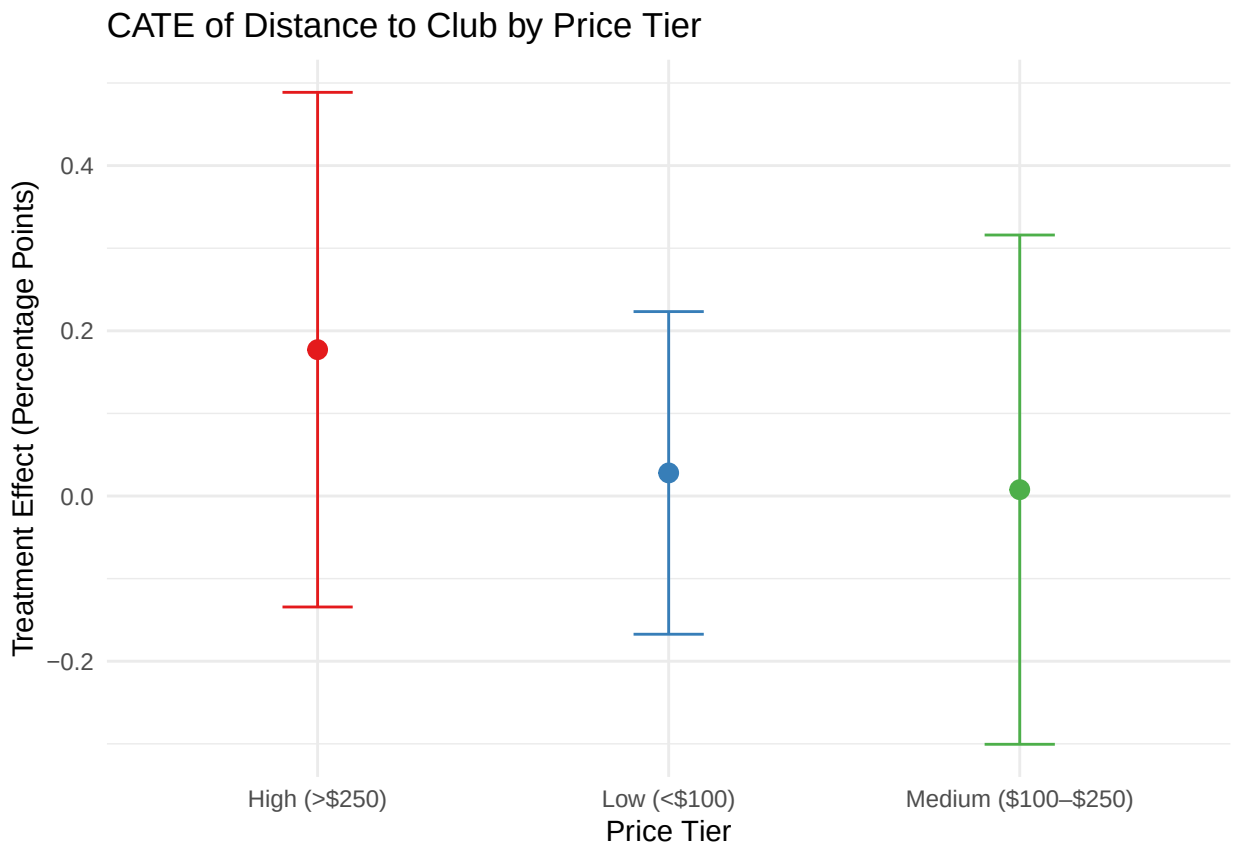


Distance to the Club

```
print(paste("CATE (All):", round(ate[1], 4), "| SE:", round(ate[2], 4)))
```

```
## [1] "CATE (All): 0.0311 | SE: 0.0822"
```

```
ggplot(cate_results, aes(x = Price_Tier, y = CATE, color = Price_Tier)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI), width = 0.2) +
  labs(
    title = "CATE of Distance to Club by Price Tier",
    x = "Price Tier",
    y = "Treatment Effect (Percentage Points)"
  ) +
  theme_minimal() +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "none")
```

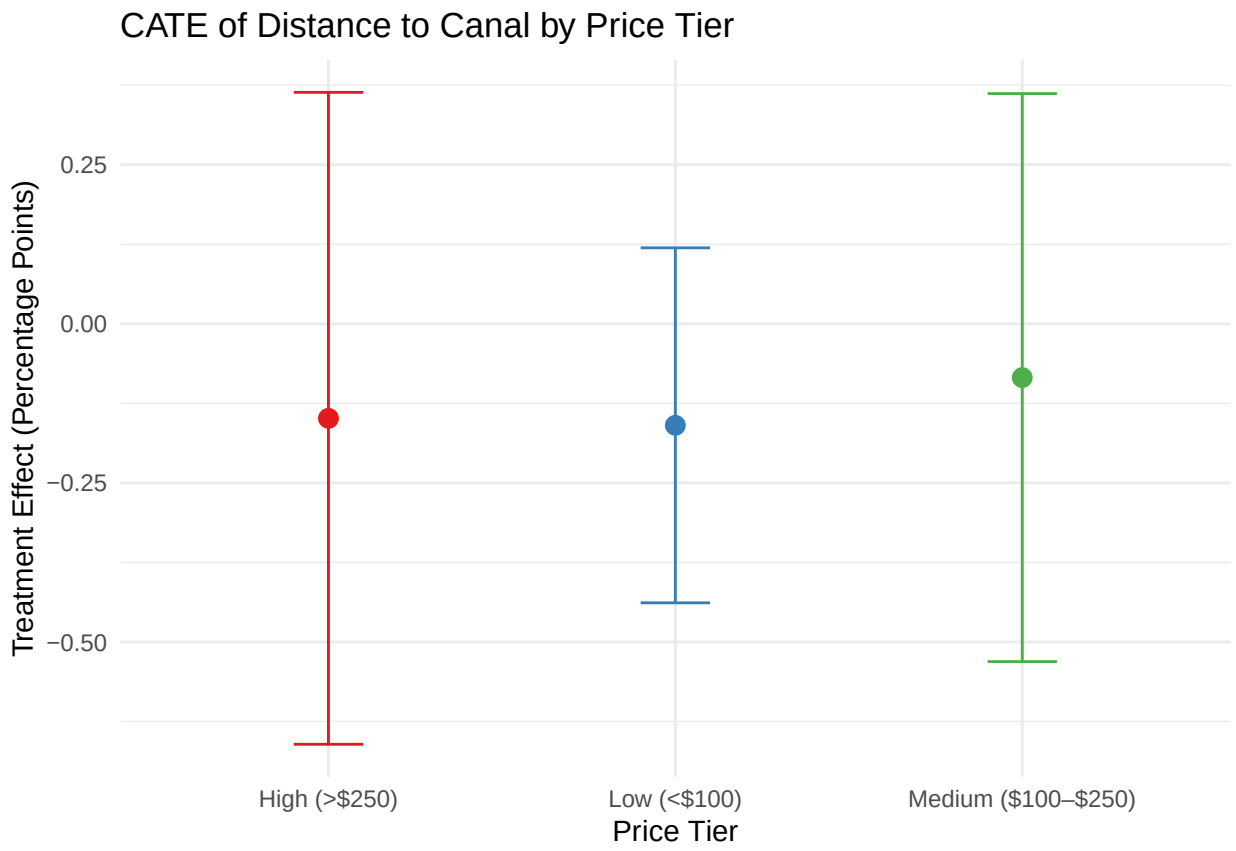


Distance to Canal

```
print(paste("CATE (All):", round(ate[1], 4), "| SE:", round(ate[2], 4)))
```

```
## [1] "CATE (All): -0.1354 | SE: 0.1175"
```

```
ggplot(cate_results, aes(x = Price_Tier, y = CATE, color = Price_Tier)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI), width = 0.2) +
  labs(
    title = "CATE of Distance to Canal by Price Tier",
    x = "Price Tier",
    y = "Treatment Effect (Percentage Points)"
  ) +
  theme_minimal() +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "none")
```

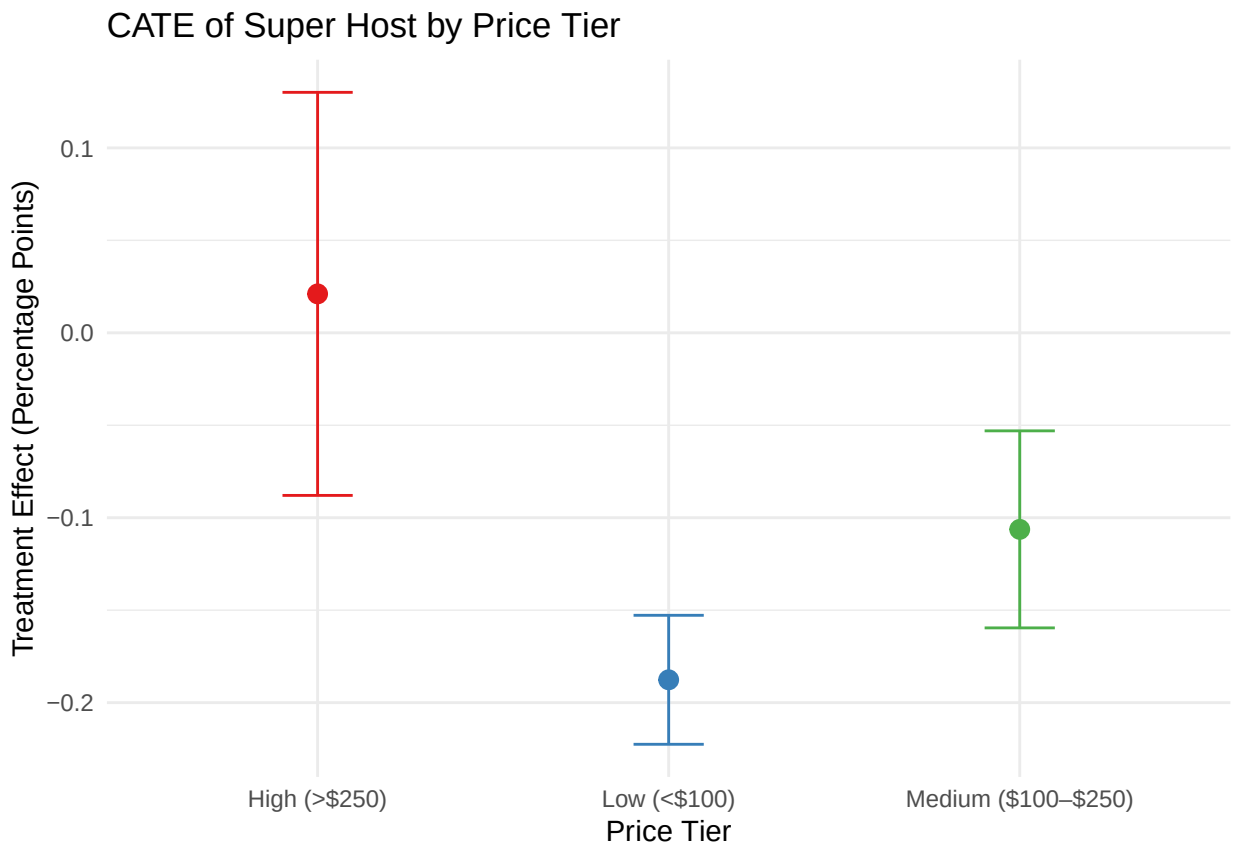


Super host

```
print(paste("CATE (All):", round(ate[1], 4), "| SE:", round(ate[2], 4)))
```

```
## [1] "CATE (All): -0.1489 | SE: 0.015"
```

```
ggplot(cate_results, aes(x = Price_Tier, y = CATE, color = Price_Tier)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI), width = 0.2) +
  labs(
    title = "CATE of Super Host by Price Tier",
    x = "Price Tier",
    y = "Treatment Effect (Percentage Points)"
  ) +
  theme_minimal() +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "none")
```



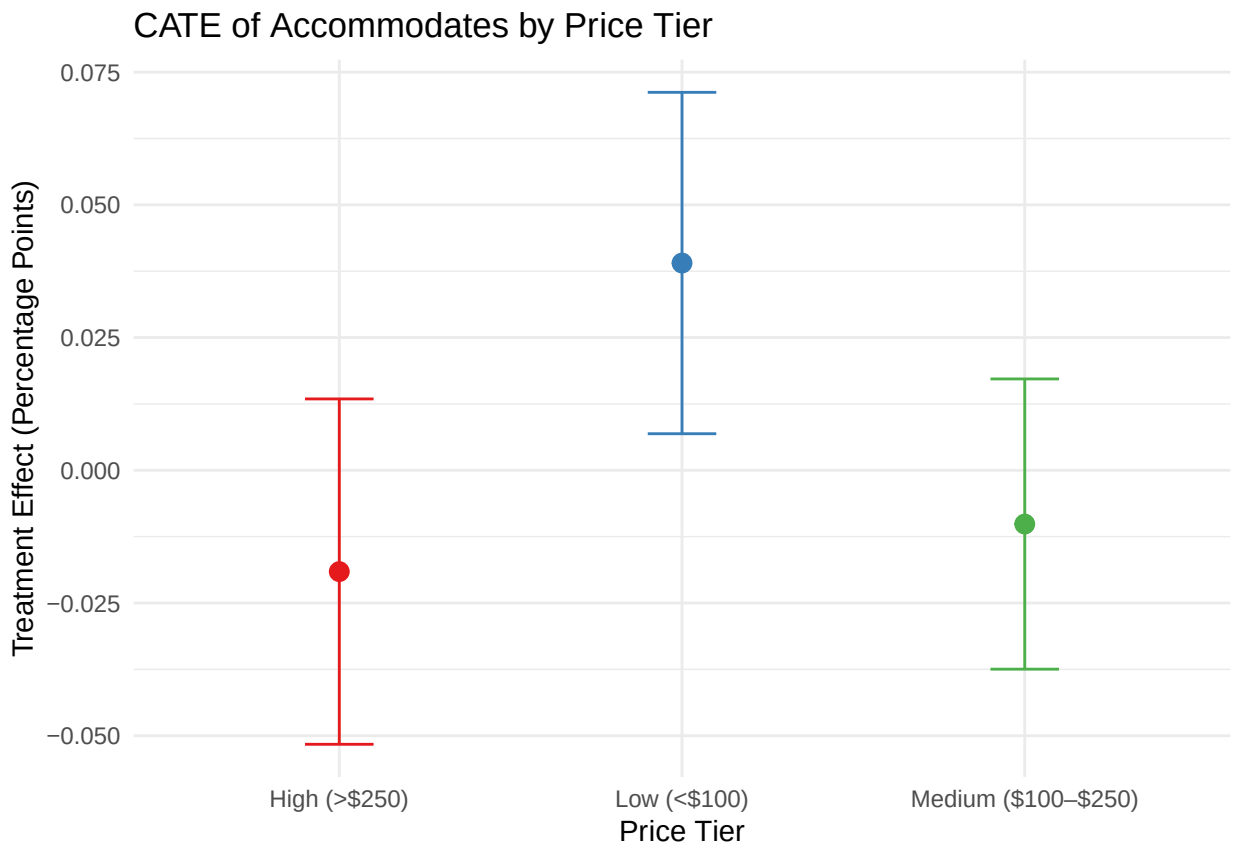
Super host status appears to have a negative effect on low cost properties. This indicates buyers looking for a cheap to medium priced place to stay do not care about super host status or can even be deterred by it

Accommodates

```
print(paste("CATE (All):", round(ate[1], 4), "| SE:", round(ate[2], 4)))
```

```
## [1] "CATE (All): 0.02 | SE: 0.0113"
```

```
ggplot(cate_results, aes(x = Price_Tier, y = CATE, color = Price_Tier)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI), width = 0.2) +
  labs(
    title = "CATE of Accommodates by Price Tier",
    x = "Price Tier",
    y = "Treatment Effect (Percentage Points)"
  ) +
  theme_minimal() +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "none")
```



For each additional person you can accommodate in a low cost apartment, your chance of achieving a high booking rate inreases by 4%.

Conclusion and Recommendations

Our analysis reveals clear, actionable insights for investors targeting Entrepôt's Airbnb market. Location is paramount—properties near Gare de Paris consistently achieve higher booking rates, while proximity to nightlife or canals shows negligible impact. For budget listings (<$100), maximizing occupancy (e.g., through larger guest capacity) is critical, while mid-tier properties benefit most from instant booking and verified host profiles. Surprisingly, "Superhost" status adds little value for cheaper units, suggesting resources are better spent on other features.

To capitalize on these findings:

Prioritize transit-adjacent properties (especially near major stations),

Tailor amenities to price tiers (capacity for budget units, convenience for mid-tier), and

Streamline bookings with instant approval and review incentives.

By combining these location-specific and attribute-driven strategies, investors can minimize risk and maximize returns in Entrepôt's dynamic short-term rental market.