

Airbnb Analytics Dashboard — User Guide

Objective:

This dashboard analyzes Airbnb listings, pricing trends, availability patterns, and projected revenue scenarios using interactive filters and simulation sliders.

How to Refresh Data

1. Open Power BI Desktop file (.pbix)
2. Go to **Home tab**
3. Click **Refresh**
4. Wait until refresh completes
5. Save the file

if report is published:

6. Go to Power BI Service
7. Select Dataset
8. Click **Refresh now**

Dashboard Navigation (“What-if”)

1. Dashboard Structure Overview

The **Airbnb – Price & Scenario Analysis Dashboard** consists of **four main pages**; each designed for a specific level of analysis:

1. **Overview**
2. **Listing Analysis**
3. **Scenario Insights**
4. **Tooltip Page**

Use the **page tabs at the bottom** of the Power BI report to navigate between these pages.

2. Page-wise Navigation & Usage

Page 1: Overview

Purpose

Provides a **high-level summary** of pricing, availability, listings, and projected revenue with interactive What-If analysis.

How to Navigate

- Click on “Overview” tab at the bottom of the report.
- This is the **default landing page**.

Key Components

- **Date Range Slider** (top-left)
- **What-If Sliders**
 - Occupancy Rate
 - Seasonal Multiplier
- **KPI Cards**
 - Average Price
 - Availability Rate
 - Total Listings
 - Projected Revenue
- **Filters**
 - Property Type
 - Room Type

How to Use What-If Sliders (Overview Page)

1) Occupancy Rate Slider

- Simulates expected booking occupancy.
- Range: 0% – 100%
- Example:
 - 40% → Conservative demand
 - 80% → High demand scenario

Increasing occupancy increases **Projected Revenue** dynamically.

2) Seasonal Multiplier Slider

- Adjusts price based on seasonality.
- Example:
 - 0.80 → Off-season pricing

- 1.20 → Peak-season pricing

This multiplier is applied to the base price across all listings.

3) Date Range Slider

- Filters analysis for a specific time.
 - All KPIs and visuals update automatically.
-

Expected Outcome

- All KPI cards and charts update instantly based on slider movement.
 - Helps simulate **best-case vs worst-case revenue scenarios**.
-

Page 2: Listing Analysis

Purpose

Provides **detailed, listing-level insights** and price distribution analysis.

How to Navigate

- Click on “**Listing Analysis**” tab at the bottom.

Key Components

- **Map Visual:** Listings by location
- **Price Distribution Histogram**
- **Detailed Table:**
 - Listing ID
 - Property Type
 - Room Type
 - Accommodates
 - Bedrooms
 - Average Price

How What-If Analysis Applies

- What-If parameters selected on the Overview page **continue to affect pricing logic**.
- Filters (Room Type / Property Type) refine listing-level insights.

Use this page to understand **which listings benefit most** from scenario changes.

Page 3: Scenario Insights

Purpose

Focuses on **scenario comparison and storytelling**.

How to Navigate

- Click on “**Scenario Insights**” tab.

Key Components

- **Scenario Price vs Average Price Comparison**
- **Projected Revenue Trend**
- **Key Narrative Panel**
 - Auto-generated insights based on current scenario selections

How to Use What-If Sliders Here

- Adjust sliders from the Overview page (or synced slicers).
- Observe:
 - Change in Scenario Price
 - Revenue trend movement
 - Automatically updated narrative insights

This page explains “**what changed and why**” using data-driven text.

Page 4: Tooltip Page

Purpose

Provides **additional contextual insights** when hovering over visuals.

How to Navigate

- This page is **not accessed directly**.

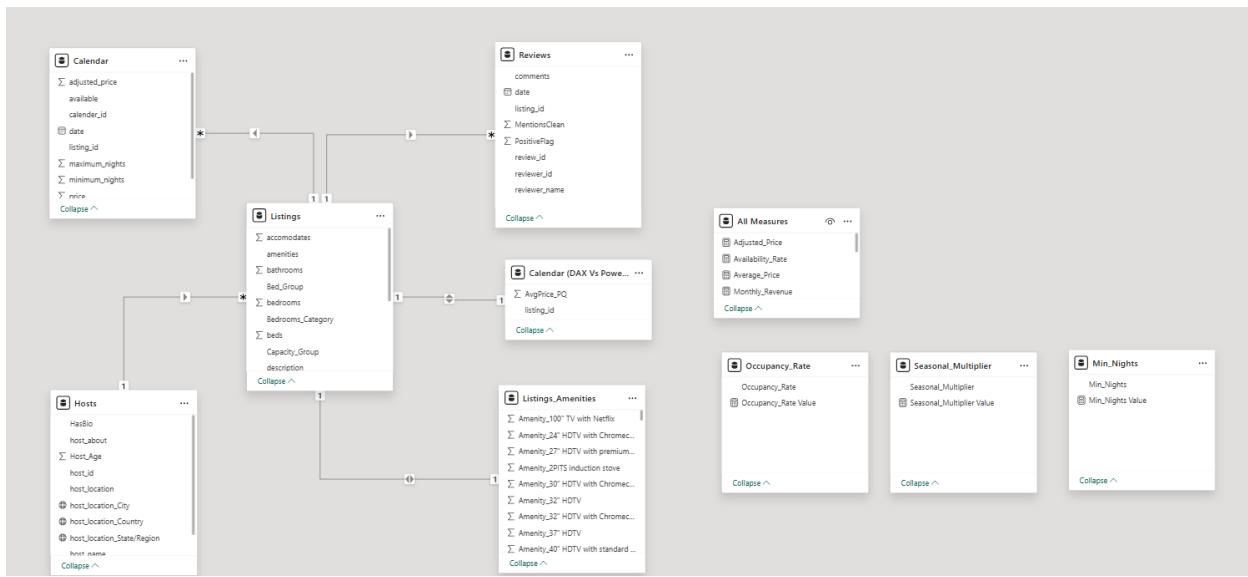
- Hover over supported visuals in other pages to see tooltip insights.

Key Metrics Displayed

- Average Price by Quarter
- Occupancy Rate Value
- Host Age
- Review Count
- Room Type Comparison
- Booked vs Available Nights

Enhances analysis without cluttering main visuals.

Data Model Diagram



List of DAX Formulas

- Adjusted_Price = [Average_Price] * SELECTEDVALUE ('Seasonal_Multiplier' [Seasonal_Multiplier])
- Availability_Rate = DIVIDE(COUNTROWS(FILTER(Calendar, Calendar[available] = TRUE())), COUNTROWS(Calendar))
- Average_Price = AVERAGE(Calendar[price])
- Monthly_Revenue = SUMX(FILTER(Calendar, Calendar[available] = TRUE()), Calendar[price])

- PositiveFlag_Review % = DIVIDE(SUM(Reviews[PositiveFlag]), COUNT(Reviews[review_id]))
- Projected_Occupied_Nights = [Total_Nights] * SELECTEDVALUE('Occupancy_Rate'[Occupancy_Rate])
- Projected_Revenue = [Adjusted_Price] * [Projected_Occupied_Nights]
- Review_Count = COUNTROWS(Reviews)
- Review_CountBy_Listing =

CALCULATE(COUNTROWS(Reviews), Reviews[listing_id] = MAX(Listings[listing_id]))
- Scenario_Price = [Average_Price] *

SELECTEDVALUE('Seasonal_Multiplier'[Seasonal_Multiplier], 1)
- Total_Active_Days =

CALCULATE(COUNT(Calendar[Date]), Calendar[available] = TRUE())
- Total_Clean_Mentions = SUM(Reviews[MentionsClean])
- Total_Listings = DISTINCTCOUNT(Listings[listing_id])
- Total_Nights = COUNTROWS(Calendar)
- Total_PositiveFlag_Reviews = SUM(Reviews[PositiveFlag])
- Min_Nights Value = SELECTEDVALUE('Min_Nights'[Min_Nights], 2)
- Occupancy_Rate Value =

SELECTEDVALUE('Occupancy_Rate'[Occupancy_Rate], 0.75)
- Seasonal_Multiplier Value =

SELECTEDVALUE('Seasonal_Multiplier'[Seasonal_Multiplier], 1)

Power Query (Key M-Code)

Hosts Table

```

#"Duplicated Column1" = Table.DuplicateColumn(#"Cleaned Text2",
"host_location", "host_location - Copy"),

#"Split Column by Delimiter1" = Table.SplitColumn(#"Duplicated Column1",
"host_location - Copy", Splitter.SplitTextByEachDelimiter({","}, QuoteStyle.Csv,
true), {"host_location - Copy.1", "host_location - Copy.2"}),

#"Changed Type3" = Table.TransformColumnTypes(#"Split Column by
Delimiter1",{{"host_location - Copy.1", type text}, {"host_location - Copy.2", type
text}}),

#"Split Column by Delimiter2" = Table.SplitColumn(#"Changed Type3",
"host_location - Copy.1", Splitter.SplitTextByEachDelimiter({","}, QuoteStyle.Csv,
true), {"host_location - Copy.1.1", "host_location - Copy.1.2"}),

```

```

#"Changed Type4" = Table.TransformColumnTypes(#"Split Column by
Delimiter2",{{"host_location - Copy.1.1", type text}, {"host_location - Copy.1.2", type
text}}),

#"Removed Columns" = Table.RemoveColumns(#"Changed Type4",{"host_location -
Country", "host_location - State/Region", "host_location - City"}),

#"Renamed Columns2" = Table.RenameColumns(#"Removed
Columns",{{"host_location - Copy.1.1", "host_location_City"}, {"host_location -
Copy.1.2", "host_location_State/Region"}, {"host_location - Copy.2",
"host_location_Country"} }),

#"Added Conditional Column" = Table.AddColumn(#"Replaced Value4", "HasBio",
each if [host_about] <> null then true else false),

#"Changed Type5" = Table.TransformColumnTypes(#"Added Conditional
Column",{{"HasBio", type logical}, {"host_id", type text}}),

#"Replaced Errors" = Table.ReplaceErrorValues(#"Changed Type5", {"host_since", null}),

#"Added Custom" = Table.AddColumn(#"Replaced Errors", "Host_Tenure_Years",
each Duration.Days(DateTime.Date(DateTime.LocalNow()) - [host_since]) / 365),

#"Inserted Rounding" = Table.AddColumn(#"Added Custom", "Round", each
Number.Round([Host_Tenure_Years], 2), type number),

#"Renamed Columns4" = Table.RenameColumns(#"Removed Columns2",{{"Round",
"Host_Age"}})

```

LISTINGS TABLE

```

#"Replaced Value" = Table.ReplaceValue(#"Changed Type","Half-bath","0.5
bath",Replacer.ReplaceText,{"bathrooms_text"}),

#"Replaced Value1" = Table.ReplaceValue(#"Replaced Value","Private half-
bath","0.5 bath",Replacer.ReplaceText,{"bathrooms_text"}),

#"Replaced Value2" = Table.ReplaceValue(#"Replaced Value1","Shared half-
bath","0.5 bath",Replacer.ReplaceText,{"bathrooms_text"}),

#"Added Custom" = Table.AddColumn(#"Replaced Value2", "bathrooms", each try
Number.FromText(Text.BeforeDelimiter([bathrooms_text], " ")) otherwise null),

#"Changed Type1" = Table.TransformColumnTypes(#"Added
Custom",{{"bathrooms", type number}}),

#"Added Custom1" = Table.AddColumn(#"Removed Columns",
"Bedrooms_Category", each if [bedrooms] = null then "Not Specified"
else if [bedrooms] = 1 then "1 BR"

```

```

else if [bedrooms] <= 3 then "2–3 BR"
else "4+ BR"),

#"Added Custom2" = Table.AddColumn(#"Cleaned Text", "Capacity_Group ", each if
[accomodates] <= 2 then "1–2 Guests" else if [accomodates] >= 4 then "3–4 Guests"
else if [accomodates] <= 6 then "5–6 Guests" else "7+ Guests"),

#"Changed Type2" = Table.TransformColumnTypes(#"Added
Custom2",{{"Capacity_Group ", type text}}),

#"Added Custom3" = Table.AddColumn(#"Changed Type2", "Bed_Group", each if
[beds] <= 1 then "1 Bed" else if [beds] <= 3 then "2–3 Beds" else "4+ Beds"),

#"Replaced Value3" = Table.ReplaceValue(#"Filtered
Rows",null,0,Replacer.ReplaceValue,{"beds"}),

#"Added Custom4" = Table.AddColumn(#"Replaced Value3", "Bed_Group 1", each if
[beds] <= 1 then "1 Bed"
else if [beds] <= 3 then "2–3 Beds"
else "4+ Beds"),

#"Changed Type3" = Table.TransformColumnTypes(#"Renamed
Columns",{{"Bed_Group", type text}})

```

REVIEWS TABLE

```

#"Added Custom" = Table.AddColumn(#"Changed Type1", "PositiveFlag", each if
[comments] = null then 0
else if Text.Contains(Text.Lower([comments]), "great")
    or Text.Contains(Text.Lower([comments]), "excellent")
    or Text.Contains(Text.Lower([comments]), "amazing")
    or Text.Contains(Text.Lower([comments]), "perfect")
then 1
else 0),

#"Added Custom1" = Table.AddColumn(#"Added Custom", "MentionsClean", each if
[comments] = null then 0
else if Text.Contains(Text.Lower([comments]), "clean")
then 1
else 0),

```

```
#"Changed Type2" = Table.TransformColumnTypes(#"Added  
Custom1",{{"PositiveFlag", Int64.Type}, {"MentionsClean", Int64.Type}})
```

In #"**Changed Type2**"

Instructions to Update What-If Parameter Tables

1. Overview

This dashboard uses **What-If parameter tables** to simulate different pricing and demand scenarios.

The parameters currently used are:

- **Occupancy Rate**
- **Seasonal Multiplier**

These tables are **disconnected tables** and control scenario calculations through DAX measures.

2. Identify Existing What-If Parameter Tables

1. Open the report in **Power BI Desktop**
2. In the **Fields pane**, locate:
 - Occupancy_Rate
 - Seasonal_Multiplier
3. Verify:
 - No relationships exist with fact tables
 - Each table has an auto-generated **Value measure** (e.g., Occupancy_Rate Value)

Do not create relationships for What-If tables.

3. Update Parameter Ranges (Recommended Method)

Power BI does **not support editing** parameter ranges after creation.
To update min/max/step values, **recreate the parameter**.

A. Update Occupancy Rate Parameter

1. Go to **Modeling** → **New Parameter** → **Numeric**
2. Configure:

- **Minimum:** 0
 - **Maximum:** 1
 - **Increment:** 0.05
 - **Default:** 0.40
3. Click **OK**
 4. Delete the old Occupancy Rate slicer
 5. Add the new slicer to the report
 6. Update DAX measures if the parameter name has changed

Always use **[Occupancy_Rate Value]** in calculations.

B. Update Seasonal Multiplier Parameter

1. Go to **Modeling** → **New Parameter** → **Numeric**
2. Configure:
 - **Minimum:** 0.5
 - **Maximum:** 2.0
 - **Increment:** 0.1
 - **Default:** 1.0
3. Replace the old slicer with the new one

Use **[Seasonal_Multiplier Value]** in all pricing measures.

4. Update DAX Measures (If required)

Ensure all scenario calculations reference **Value measures only**.

Example: Projected Revenue

Projected Revenue =

```
SUMX (  
    'calendar',  
    'calendar'[price]  
    * [Seasonal_Multiplier Value] * [Occupancy_Rate Value])
```