Airbnb Market Affordability Analysis

Out analysis examines the relative affordability of four key Hawaiian markets – Maui, Kauai, The Big Island, and Oahu – for two distinct traveler segments: solo travelers and families. Our findings reveal that for solo travelers, the Big Island is the most affordable market, followed by Maui and Oahu, with Kauai being the least affordable. For families, The Big Island also ranked as the most affordable market, while Oahu and Maui followed closely, with Kauai remaining less affordable.

To arrive at our findings, we first quantified what it meant for a market to be affordable. The first thing we did was to segment and understand the needs of solo and families travelers. We defined solo travelers to be groups of no more than 2 people, while family travelers are groups of no less than 4 people. After that, we developed a scoring system that considers several key factors, including the effective average nightly price, amenities, and seasonality. For solo travelers, we calculate the presence of Wi-Fi and Laptop-friendly workspace and for family travelers, we counted the number of cribs, highchairs, Pack 'n Play, and washers

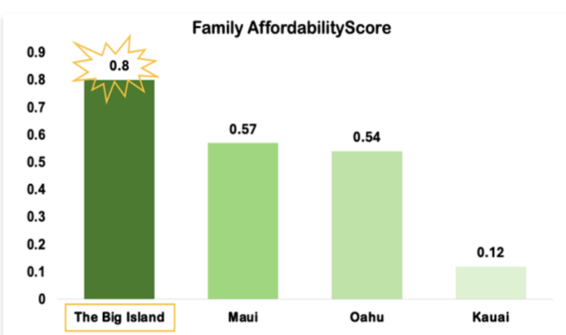
Our methodology is as follows

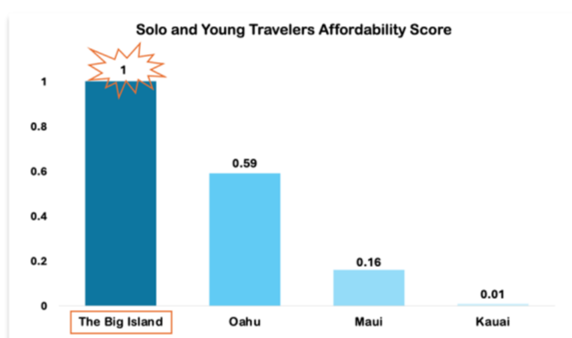
- We first ranked each of the three categories to get an overall score of how each of the for markets ranked for solo travelers and families.
- We then normalized the data to ensure that the different scales of the metrics would be comparable across pricing, amenities and seasonality.

After that, we created an affordability score system to evaluates and compare the four markets. This system is as follows:

$$Affordabiliy\ score = 0.6 \times Price_{normalized} + 0.2 \times Amenity_{normalized} + 0.2 \times Seasonality_{normalized}$$

We expect effective price to be the predominant factor affecting affordability, and thus it was given higher weighting compared to the other two factors. By effective price we mean the price with all the additional fees included (cleaning, security deposit, extra guests, etc.)





The Big Island's consistent affordability for both solo and family travelers make it a budget friendly destination. Airbnb can highlight this in its marketing campaigns to incentivize host and attract more visitors. Maui and Oahu strike balances affordability and amenities, which makes these markets a good choice for travelers who want to get an unforgettable experience without breaking the bank.

Airbnb Market Affordability Analysis
Winning Performance Analysis

The analysis reveals that the significance of 3-point field goals increased notably in the 2000s (2000–2009) compared to the 1990s (1990–1999). Specifically, teams that scored a higher proportion of their total points from 3-pointers were more likely to win games in the 2000s than in the 1990s.

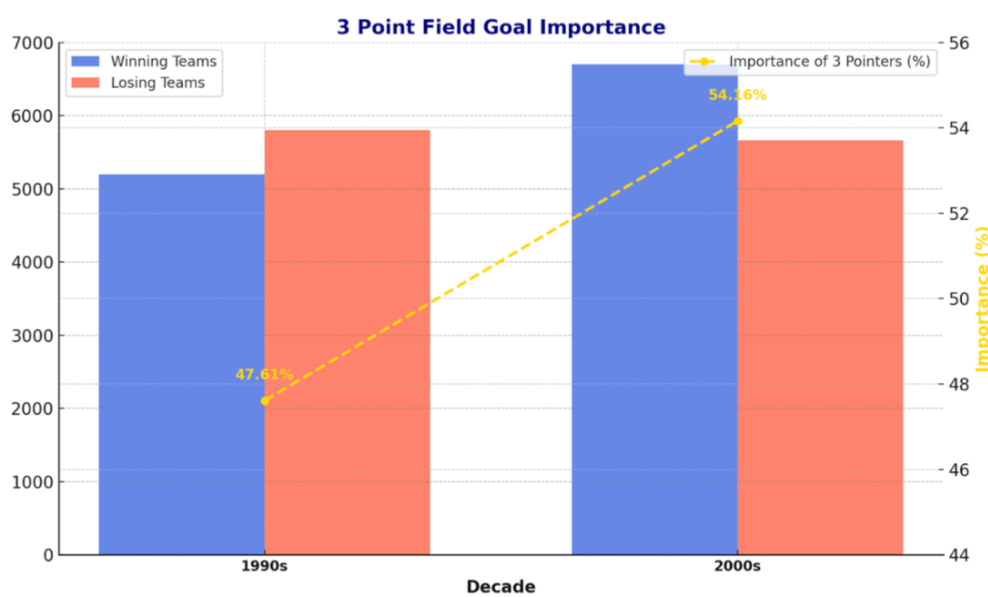
To assess this, we used the following key metrics:

1. **Won with More 3P**: The number of winning teams in each time period that had a higher percentage of their points from 3-pointers compared to the losing team. (Defined as **"_id: true"** in MangoDB)
2. **Lost with More 3P**: The number of losing teams in each period that had a higher percentage of points from 3-pointers compared to the winning team. (Defined as **"_id: false"** in MangoDB)
3. **Importance of 3P**: Defined as the **percentage** of games won by teams with a greater contribution of 3-pointers to their total score.

$$\textbf{Percentage} = \frac{\textbf{Won with More 3P}}{\textbf{Won with More 3P} + \textbf{Lost with More 3P}} \times 100\%$$

$$\text{EX:}\quad \text{Percentage in 2000s} = \frac{6641}{5620 + 6641} \times 100\%$$

The data shows that **47.61%** of teams in the 1990s won games with a higher 3-point contribution, compared to **54.16%** in the 2000s — a 6**.56% increase**. To validate this, I conducted a proportion z-test, confirming with 99% confidence that the importance of 3-point field goals as a predictor of winning has significantly risen between these two decades.



Appendix for Airbnb dataset

```
db = db.getSiblingDB("sample_airbnb");
db.listingsAndReviews.findOne();
//solo
db.listingsAndReviews.aggregate([
  { $match: { "address.market": { $in: ["Maui", "Kauai", "The Big Island", "Oahu"] }, accommodates: {
$lte: 2 }}},
  { $project: {
     market: "$address.market",
     soloPrice: {
        $add: [
           { $toDouble: "$price" },
           { $divide: [ { $add: [{ $toDouble: "$cleaning_fee" }, { $toDouble: "$security_deposit" }] }, 2 ] }
        ]
     },
     amenityScore: {
        $sum: [
           { $cond: [ { $in: ["Wifi", "$amenities"] }, 1, 0 ] },
           { $cond: [ { $in: ["Laptop friendly workspace", "$amenities"] }, 1, 0 ] }
        ]
     },
     avgSeasonality: "$availability.availability_365"
  }},
  { $group: {
     _id: "$market",
     avgSoloPrice: { $avg: "$soloPrice" },
     amenityScore: { $avg: "$amenityScore" },
     avgSeasonality: { $avg: "$avgSeasonality" }
  }},
  { $sort: { avgSoloPrice: 1 }}
]);
//family
db.listingsAndReviews.aggregate([
  { $match: { "address.market": { $in: ["Maui", "Kauai", "The Big Island", "Oahu"] }, accommodates: {
$gte: 4 }}},
  { $project: {
     market: "$address.market",
     familyPrice: {
        $add: [
           { $toDouble: "$price" },
           { $cond: [ { $gt: [4, { $toDouble: "$guests_included" }] }, { $multiply: [{ $toDouble:
"$extra_people" }, { $subtract: [4, { $toDouble: "$guests_included" }] }] }, 0 ] },
           { $divide: [ { $add: [{ $toDouble: "$cleaning_fee" }, { $toDouble: "$security_deposit" }] }, 4 ] }
        ]
     },
     amenityScore: {
        $sum: [
           { $cond: [ { $in: ["Crib", "$amenities"] }, 1, 0 ] },
           { $cond: [ { $in: ["High chair", "$amenities"] }, 1, 0 ] },
           { $cond: [ { $in: ["Pack ’n Play/travel crib", "$amenities"] }, 1, 0 ] },
```

```
            { $cond: [ { $in: ["Washer", "$amenities"] }, 1, 0 ] }
          ]
        },
        avgSeasonality: "$availability.availability_365"
    }},
    { $group: {
        _id: "$market",
        avgFamilyPrice: { $avg: "$familyPrice" },
        amenityScore: { $avg: "$amenityScore" },
        avgSeasonality: { $avg: "$avgSeasonality" }
    }},
    { $sort: { avgFamilyPrice: 1 }}
]);
```

Python Code

```
import pandas as pd

# Solo Travelers
solo_data = [
    {"_id": "The Big Island", "avgSoloPrice": 202.22727272727272, "amenityScore": 1.7,
"avgSeasonality": 223.86},
    {"_id": "Oahu", "avgSoloPrice": 222.13492063492063, "amenityScore": 1.6785714285714286,
"avgSeasonality": 202.03571428571428},
    {"_id": "Kauai", "avgSoloPrice": 257.53333333333336, "amenityScore": 1.3333333333333333,
"avgSeasonality": 200.27777777777777},
    {"_id": "Maui", "avgSoloPrice": 258.39473684210526, "amenityScore": 1.6206896551724137,
"avgSeasonality": 201.06896551724137}
]

# Families
family_data = [
    {"_id": "The Big Island", "avgFamilyPrice": 363.375, "amenityScore": 1.1566265060240963,
"avgSeasonality": 226.24096385542168},
    {"_id": "Maui", "avgFamilyPrice": 409.78735632183907, "amenityScore": 1.2347826086956522,
"avgSeasonality": 204.5391304347826},
    {"_id": "Oahu", "avgFamilyPrice": 447.7814960629921, "amenityScore": 1.3706293706293706,
"avgSeasonality": 202.24475524475525},
    {"_id": "Kauai", "avgFamilyPrice": 523.9256756756756, "amenityScore": 1.288888888888889,
"avgSeasonality": 193.2}
]

# Normalize function
def normalize(df, column):
    min_val = df[column].min()
    max_val = df[column].max()
    return (df[column] - min_val) / (max_val - min_val)

# Affordability Scores for Solo Travelers
solo_df = pd.DataFrame(solo_data)
solo_df["market"] = solo_df["_id"]
```

```
solo_df["normalizedPrice"] = 1 - normalize(solo_df, "avgSoloPrice")
solo_df["normalizedAmenity"] = normalize(solo_df, "amenityScore")
solo_df["normalizedSeasonality"] = normalize(solo_df, "avgSeasonality")
solo_df["affordabilityScore"] = (
    0.6 * solo_df["normalizedPrice"] +
    0.2 * solo_df["normalizedAmenity"] +
    0.2 * solo_df["normalizedSeasonality"]
)

# Affordability Scores for Families
family_df = pd.DataFrame(family_data)
family_df["market"] = family_df["_id"]
family_df["normalizedPrice"] = 1 - normalize(family_df, "avgFamilyPrice")
family_df["normalizedAmenity"] = normalize(family_df, "amenityScore")
family_df["normalizedSeasonality"] = normalize(family_df, "avgSeasonality")
family_df["affordabilityScore"] = (
    0.6 * family_df["normalizedPrice"] +
    0.2 * family_df["normalizedAmenity"] +
    0.2 * family_df["normalizedSeasonality"]
)

# Save CSV files
solo_df[["market", "avgSoloPrice", "amenityScore", "avgSeasonality",
"affordabilityScore"]].to_csv("solo_affordability_scores_with_market.csv", index=False)
family_df[["market", "avgFamilyPrice", "amenityScore", "avgSeasonality",
"affordabilityScore"]].rename(
    columns={"avgFamilyPrice": "avgSoloPrice"}
).to_csv("family_affordability_scores_with_market.csv", index=False)
```

Appendix for Airbnb dataset

```
//1990's
db.games.aggregate([
  {
   $match: {
     date: {
            $gte: ISODate("1990-01-01T00:00:00.000Z"),
            $lte: ISODate("1999-12-31T23:59:59.999Z")
     }
   }
  },
  {
   $unwind: '$box'
  },
  {
   $project: {
     _id: '$_id',
     won: '$box.won',
     fg3: '$box.team.fg3',
     totalScore: '$box.team.pts',
     fg3Percentage: {
            $cond: [
            { $gt: ['$box.team.pts', 0] }, // Avoid division by zero
            { $divide: [{ $multiply: ['$box.team.fg3', 3] }, '$box.team.pts'] },
            0
            ]
     }
   }
  },
  {
   $group: {
     _id: '$_id',
     fg3WinningPercentage: {
            $sum: {
            $cond: [{ $eq: ['$won', 1] }, '$fg3Percentage', 0]
            }
     },
     fg3LosingPercentage: {
            $sum: {
            $cond: [{ $eq: ['$won', 0] }, '$fg3Percentage', 0]
            }
     }
   }
  },
  {
   $project: {
     _id: '$_id',
     winningTeamHigherPercentage: {
            $gt: ['$fg3WinningPercentage', '$fg3LosingPercentage']
     }
```

```
    }
  },
  {
    $group: {
      _id: '$winningTeamHigherPercentage',
      count: { $sum: 1 }
    }
  }
]);

//2000's
db.games.aggregate([
  {
    $match: {
      date: {
        $gte: ISODate("2000-01-01T00:00:00.000Z"),
        $lte: ISODate("2009-12-31T23:59:59.999Z")
      }
    }
  },
  {
    $unwind: '$box'
  },
  {
    $project: {
      _id: '$_id',
      won: '$box.won',
      fg3: '$box.team.fg3',
      totalScore: '$box.team.pts',
      fg3Percentage: {
        $cond: [
        { $gt: ['$box.team.pts', 0] }, // Avoid division by zero
        { $divide: [{ $multiply: ['$box.team.fg3', 3] }, '$box.team.pts'] },
        0
        ]
      }
    }
  },
  {
    $group: {
      _id: '$_id',
      fg3WinningPercentage: {
        $sum: {
        $cond: [{ $eq: ['$won', 1] }, '$fg3Percentage', 0]
        }
      },
      fg3LosingPercentage: {
        $sum: {
        $cond: [{ $eq: ['$won', 0] }, '$fg3Percentage', 0]
        }
      }
```

```
    }
  },
  {
    $project: {
      _id: '$_id',
      winningTeamHigherPercentage: {
            $gt: ['$fg3WinningPercentage', '$fg3LosingPercentage']
      }
    }
  },
  {
    $group: {
      _id: '$winningTeamHigherPercentage',
      count: { $sum: 1 }
    }
  }
]);
```

Python Code used for the hypothesis testing:

```
from statsmodels.stats.proportion import proportions_ztest
# Input data
p1 = 0.4761  # Proportion for 1990s
p2 = 0.5416  # Proportion for 2000s
n1 = 10982  # Total games in the 1990s
n2 = 12261  # Total games in the 2000s
# Calculate number of successes
x1 = p1 * n1
x2 = p2 * n2
# Perform z-test
stat, p_value = proportions_ztest([x1, x2], [n1, n2])
# Print results
print(f"Z-Statistic: {stat:.3f}")
print(f"P-Value: {p_value:.3f}")
if p_value < 0.01:
    print("The difference is statistically significant.")
else:
    print("The difference is not statistically significant.")
```