Question 1: Airbnb dataset: *Use the data available in the Airbnb dataset to assess which of the Airbnb rental markets covered by this dataset is "the most affordable".*

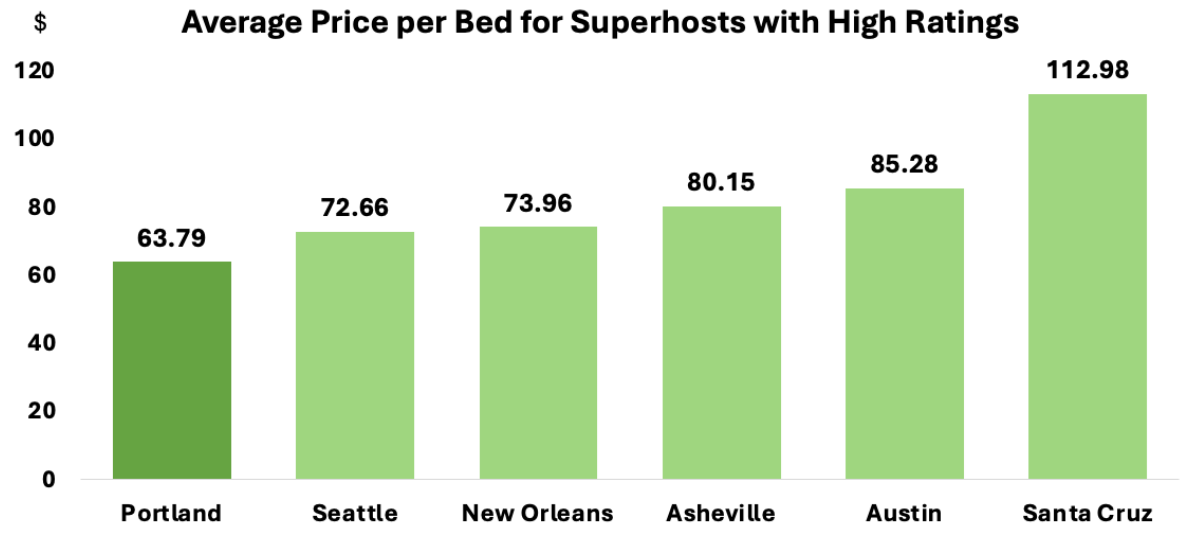
AIRBNB affordable markets Metric
1. Price per Bed for Superhosts with High Ratings

Metrics Selection
To determine "affordability," we defined it as the "Average Price per Bed for Superhosts with High Ratings" for Airbnb listings that meet the following conditions:

- MetroArea: Every Metro Area
- Superhost status: Only listings hosted by Superhosts.
- High review rating: Listings with a rating above 90, ensuring quality and reliability.
- At least one bed: Ensures comparability across listings.
- To calculate the "Average Price per Bed" for each city, we used the formula:
  Price per Bed = Listing Price / Number of Beds

This metric aligns with the typical preferences for affordable and well-served accommodations when choosing Airbnb rentals for trips with friends or family. The "Average Price per Bed" provides a standardized way to compare costs across cities with similar high-quality conditions.

Results



The bar chart provides a visual representation of these results, clearly showing that **Portland** has the lowest average price per bed among the analyzed cities, making it the most affordable option for quality Airbnb listings with Superhosts.

Conclusion
Portland is **the most affordable Airbnb rental market** among the cities analyzed**,** with an average price per bed of $63.79 for listings with Superhosts and ratings above 90.

Question 2: Bikeshare dataset: *Select two metro areas from the Bikeshare dataset and assess in which of those metro areas the bike rental business is "most affected" by weather.*

Selection of two cities: Los Angeles and Portland:
We chose Los Angeles and Portland to compare the impact of weather on bike rentals in a city with stable weather, like LA, versus one with more rain and a varied climate, like Portland.

Explanation of Quantitative Metrics:

Weather Category Metric: We began our analysis by redefining what it means to be "affected by weather." For instance, we agreed that this impact can be measured by comparing the total number of trips during good weather with those during bad weather. By analyzing the decrease in trip usage under bad weather for each city, we could determine which city was more affected.

Next, we defined "good" and "bad" weather conditions. We analyzed several features from the weather table in the BikeShare database, set specific thresholds for each condition, and created a weighted scoring system to classify each period as having good or bad weather. This approach gave us a criterion under which we can assess weather conditions across multiple factors in both cities. The table below will explain our logic and scoring system.

Weather Feature	Condition for Points	Points Added
Temperature (Temp_f)	Below 40°F or above 95°F	2
Feels Like Temperature (FeelsLike_f)	Below 40°F or above 95°F	2
Minimum Temperature (Temp_min_f)	Below 40°F or above 95°F	2
Maximum Temperature (Temp_max_f)	Below 40°F or above 95°F	2
Wind Speed (WindSpeed_mph)	Above 20 mph	1.5
Rain (1-hour, Rain_mm_1h)	Any measurable rain	3
Rain (3-hour, Rain_mm_3h)	Any measurable rain	3
Relative Humidity	Above 80% or below 20%	0.5
Cloud Cover (PercentCloudCover)	Above 80%	0.5
Snow (1-hour, Snow_mm_1h)	Any measurable snow	3
Snow (3-hour, Snow_mm_3h)	Any measurable snow	3

Good Weather: Score below 5 | **Bad Weather**: Score of 5 or higher

Percentage decline in rider Usage Metric: Finally, we aggregated our results by city and time, calculating the total number of rides during both good and bad weather conditions for each city. We then computed the percentage decrease in total rides for bad weather conditions. By comparing the percentage decline in each city, we identified that Los Angeles was most affected by bad weather conditions.

MetroID	City	Bad Weather Trips	Good Weather Trips	Percentage Decrease
2	Los Angeles	4,386.00	1,097,682.00	99.60%
4	Portland	102,374.00	1,182,538.00	91.34%

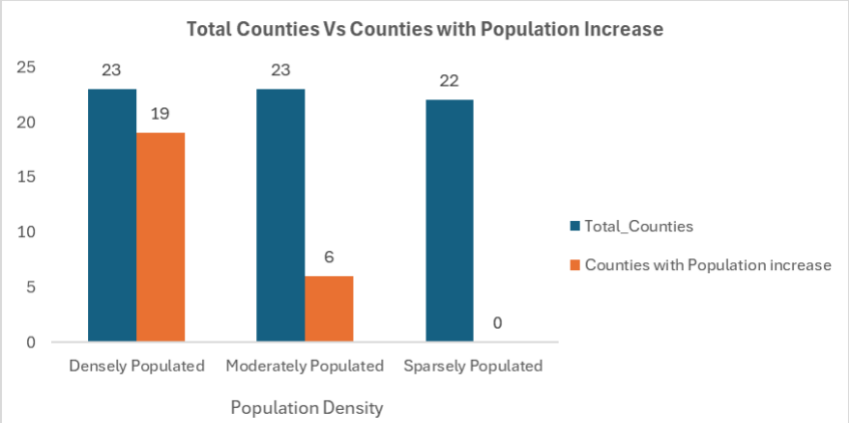
Conclusion:
Among both cities analyzed, Los Angeles was the most affected by weather because it showed a greater decrease in bike rentals during bad weather conditions compared to Portland, 99.60% vs 91.34%

Question 3: US Demographics dataset: assess the relative change in population in counties in the state of Pennsylvania between 2011 and 2021.

Metrics: Standards and Measurements for Classifying Population Density
To categorize counties as densely, moderately, or sparsely populated, the population density of each county in 2021 is determined using the subsequent formula:
Population Density = Total Population / Area in Sq. Miles

After analyzing the density distribution throughout Pennsylvania's counties, these counties are categorized into three groups based on percentiles; which we calculated using the ntile function in MySql:

- Densely populated: Counties that fall in the top one-third with a population density percentile rank exceeding 66.67%.
- Moderately populated: Counties that fall within a population density percentile rank of 33.33% to 66.67% (the central third). This equates to roughly between 83 and 227 individuals per square mile or higher.
- Sparsely inhabited: Counties that have a population density percentile ranking below 33.33% (lowest one-third). This is approximately equal to 83 individuals per square mile or greater.



Density Category	Average Population Change
Densely Populated	3.35%
Moderately Populated	-2.13%
Sparsely Populated	-5.12%

Conclusion:
Our results show that relative population changes (in percentages) between 2011 and 2021 are not consistent across the three density categories in Pennsylvania. Densely populated counties experienced the most density increase with a growth rate of 3.35%. Moderately populated cities saw a decrease in density of 2.13, and sparsely populated counties dropped by 5.12%. This comparison shows an uneven density growth across county types, signaling growth in urban areas.

Appendix

Question 1 Code:

```
SELECT
   MetroArea.CoreCity AS "City",
   ROUND(AVG(Listing.Price / Listing.Beds),2) AS "Avg_Price_per_Bed",
   Host.IsSuperhost AS "SuperHost"
FROM
   Listing
JOIN
   MetroArea ON Listing.MetroID = MetroArea.MetroID
JOIN
   Host ON Listing.HostID = Host.HostID
WHERE
   Host.IsSuperhost = 1
   AND Listing.ReviewScoresRating > 90
   AND Listing.Beds > 0
GROUP BY
   MetroArea.CoreCity, Host.IsSuperhost
ORDER BY
   "Avg_Price_per_Bed" ASC;
```

Question 2 Code:

```
WITH RentalsWeather AS (
   -- Join Rentals and Weather for all records
   SELECT
      r.MetroID,
      r.TripID,
      r.StartDate,
      r.StartTime,
      r.UserType,
      r.TripDuration_Minutes,
      w.DateOfReading,
      w.HourOfReading,
      w.Temp_f,
      w.FeelsLike_f,
      w.Temp_min_f,
      w.Temp_max_f,
      w.Barometer_mmHg,
      w.RelativeHumidity,
      w.WindSpeed_mph,
      w.Rain_mm_1h,
      w.Rain_mm_3h,
      w.PercentCloudCover,
      w.Snow_mm_1h,
      w.Snow_mm_3h
   FROM
      Rentals r
   LEFT JOIN
      Weather w
   ON
      r.MetroID = w.MetroID
      AND r.StartDate = w.DateOfReading
      AND HOUR(r.StartTime) = w.HourOfReading
),

RentalWeatherClassification AS (
   -- Calculate BadWeatherScore and classify weather for each rental
   SELECT
      MetroID,
      TripID,
      StartDate,
      StartTime,

      -- Calculate BadWeatherScore based on thresholds
      (
         (CASE WHEN Temp_f < 40 OR Temp_f > 95 THEN 2 ELSE 0 END) +
         (CASE WHEN FeelsLike_f < 40 OR FeelsLike_f > 95 THEN 2 ELSE 0 END) +
         (CASE WHEN Temp_min_f < 40 OR Temp_min_f > 95 THEN 2 ELSE 0 END) +
         (CASE WHEN Temp_max_f < 40 OR Temp_max_f > 95 THEN 2 ELSE 0 END) +
         (CASE WHEN WindSpeed_mph > 20 THEN 1.5 ELSE 0 END) +
         (CASE WHEN Rain_mm_1h > 0 THEN 3 ELSE 0 END) +
```

```
        (CASE WHEN Rain_mm_3h > 0 THEN 3 ELSE 0 END) +
        (CASE WHEN RelativeHumidity > 80 OR RelativeHumidity < 20 THEN 0.5 ELSE 0 END) +
        (CASE WHEN PercentCloudCover > 80 THEN 0.5 ELSE 0 END) +
        (CASE WHEN Snow_mm_1h > 0 THEN 3 ELSE 0 END) +
        (CASE WHEN Snow_mm_3h > 0 THEN 3 ELSE 0 END)
    ) AS BadWeatherScore,

    -- Classify weather as Good or Bad
    CASE
        WHEN (
            (CASE WHEN Temp_f < 40 OR Temp_f > 95 THEN 2 ELSE 0 END) +
            (CASE WHEN FeelsLike_f < 40 OR FeelsLike_f > 95 THEN 2 ELSE 0 END) +
            (CASE WHEN Temp_min_f < 40 OR Temp_min_f > 95 THEN 2 ELSE 0 END) +
            (CASE WHEN Temp_max_f < 40 OR Temp_max_f > 95 THEN 2 ELSE 0 END) +
            (CASE WHEN WindSpeed_mph > 20 THEN 1.5 ELSE 0 END) +
            (CASE WHEN Rain_mm_1h > 0 THEN 3 ELSE 0 END) +
            (CASE WHEN Rain_mm_3h > 0 THEN 3 ELSE 0 END) +
            (CASE WHEN RelativeHumidity > 80 OR RelativeHumidity < 20 THEN 0.5 ELSE 0 END) +
            (CASE WHEN PercentCloudCover > 80 THEN 0.5 ELSE 0 END) +
            (CASE WHEN Snow_mm_1h > 0 THEN 3 ELSE 0 END) +
            (CASE WHEN Snow_mm_3h > 0 THEN 3 ELSE 0 END)
        ) >= 5
        THEN 'Bad'
        ELSE 'Good'
    END AS OverallWeather
  FROM
    RentalsWeather
)

-- Final Aggregation for MetroID 2 and 4 with City Names
SELECT
  rw.MetroID,
  ma.CoreCity AS City,
  COUNT(CASE WHEN rw.OverallWeather = 'Bad' THEN 1 END) AS Total_Bad_Weather_Trips,
  COUNT(CASE WHEN rw.OverallWeather = 'Good' THEN 1 END) AS Total_Good_Weather_Trips,
  ROUND(
    (COUNT(CASE WHEN rw.OverallWeather = 'Good' THEN 1 END) - COUNT(CASE WHEN
rw.OverallWeather = 'Bad' THEN 1 END))
    / NULLIF(COUNT(CASE WHEN rw.OverallWeather = 'Good' THEN 1 END), 0) * 100, 2
  ) AS Percentage_Decrease
FROM
  RentalWeatherClassification rw
JOIN
  MetroArea ma ON rw.MetroID = ma.MetroID
WHERE
  rw.MetroID IN (2, 4)
GROUP BY
  rw.MetroID, ma.CoreCity
ORDER BY
  Percentage_Decrease DESC;
```

Question 3 Code:

```
CREATE TEMPORARY TABLE u4 as
SELECT
   county,
   state,
   Pop_2011,
   Pop_2021,
   (Pop_2021 / total_area) as PopPerSq_2021,
   (Pop_2011/total_area) AS PopPerSq_2011,
    ROUND(((Pop_2021 - Pop_2011) * 1.0 / Pop_2011) * 100, 2)
   AS pop_relative_change
FROM
   (SELECT
      L.county,
      L.state,
      SUM(E.pop2011) AS Pop_2011,
      SUM(E.pop2021) AS Pop_2021,
     sum(L.area_land_sq_miles) as total_area
   FROM
      location L
      LEFT JOIN acs_estimates E ON L.zip = E.zip
   WHERE
      L.state = 'PA'
   GROUP BY
      L.county
   ) AS county_data
ORDER BY
   PopPerSq_2021 DESC;


 create temporary table u5 as
  SELECT
   county,
   PopPerSq_2021,
   PopPerSq_2011,
   State,
   pop_relative_change,
   CASE
      WHEN ntile_bucket = 1 THEN 'densely populated'
      WHEN ntile_bucket = 2 THEN 'moderately populated'
      ELSE 'sparsely populated'
   END AS density_category
FROM
   (SELECT
      county,
      PopPerSq_2021,
      PopPerSq_2011,
      state,
      pop_relative_change,
      NTILE(3) OVER (ORDER BY PopPerSq_2021 DESC) AS ntile_bucket
```

```
    FROM
        u4
    ) AS temp_with;

SELECT
/*Count(*) as Total_Counties, sum(case when pop_relative_change>0 then 1 else 0 end) as "Counties
with Population increase",
density_category , */density_category,
round(avg(pop_relative_change),2) as avg_pop_change
from u5
group by density_category;
```