Question 1: What is the most common room type in NYC Airbnb listings?

SELECT room\_type, COUNT(\*)

FROM room\_types

GROUP BY 1

ORDER BY 2 DESC;

Question 2: What is the average price of a listing by room type?

SELECT room\_type, ROUND(AVG(price),2) avg\_price\_room

FROM prices AS p

JOIN room\_types AS r

ON p.listing\_id=r.listing\_id

GROUP BY room\_type

ORDER BY 2 DESC;

Question 3: Which borough has the highest average price per month?

SELECT borough, ROUND(AVG(price\_per\_month)::decimal, 2) AS avg\_price

FROM prices

GROUP BY borough

ORDER BY avg\_price DESC;

Question 4: How many listings of each room type are in each borough?

SELECT borough, room\_type, COUNT(p.listing\_id)

FROM prices AS p

JOIN room\_types AS r

ON r.listing\_id=p.listing\_id

GROUP BY 1,2

ORDER BY 1 ASC;

Question 5: How many listings in each room type category have a price of over $500 per night?

SELECT room\_type, COUNT(\*)

FROM prices AS p

JOIN room\_types AS r

ON r.listing\_id=p.listing\_id

WHERE price>500

GROUP BY 1

ORDER BY 2 DESC;

Question 6: What is the distribution of listing prices by neighborhood?

SELECT borough, MIN(price) min\_price, MAX(price) Max\_price, ROUND(AVG (price),2) avg\_price

FROM prices

GROUP BY 1;

Question 7: What is the estimated amount of revenue generated by hosts in each borough?

SELECT borough, SUM(price\*booked\_days\_365)

FROM prices

JOIN reviews

USING(listing\_id)

GROUP BY 1

ORDER BY 2 DESC;

Question 8: What is the average price per month for listings in each neighborhood?

SELECT neighbourhood,room\_type, ROUND(AVG(price\_per\_month)::decimal,2) avg\_price\_month

FROM prices

JOIN room\_types

USING (listing\_id)

GROUP BY 1,2

ORDER BY 3 DESC;

Question 9: How many listings have no reviews?

SELECT COUNT(listing\_id)

FROM reviews

WHERE number\_of\_reviews=0;

Question 10: How do the estimated book days correlate with the price of an Airbnb listing in New York City?

SELECT corr(booked\_days\_365,price) AS correlation

from prices

JOIN reviews

USING (listing\_id);

Bonus Question 1: What is the average price per room type for listings that have at least 100 reviews and are available more than 200 days a year?

SELECT room\_type, ROUND(AVG(price),2) avg\_price

FROM prices

JOIN reviews

USING (listing\_id)

JOIN room\_types

USING (listing\_id)

WHERE number\_of\_reviews>=100 AND availability\_365>200

GROUP BY 1

ORDER BY 2 DESC;

Bonus Question 2: How many hosts have more than one listing, and whats the maximum number of listings by a single host name?

SELECT host\_name, COUNT(listing\_id)

FROM reviews

GROUP BY 1

ORDER BY 2 DESC;

Bonus Question 2: How many hosts have more than one listing?

WITH t1 AS(SELECT host\_name, COUNT(listing\_id)

FROM reviews

GROUP BY 1

HAVING COUNT(listing\_id)>1

ORDER BY 2 DESC)

SELECT COUNT(host\_name)

FROM t1

 Bonus Question 2.1: whats the maximum number of listings by a single host name?

 SELECT host\_name, COUNT(listing\_id)

FROM reviews

GROUP BY 1

HAVING COUNT(listing\_id)>1

ORDER BY 2 DESC

LIMIT 1;

Bonus Question 3: Determine the top 5 hosts who have the highest price\_per\_month for their listings, considering only hosts who have at least 10 listings.

SELECT host\_name, COUNT(reviews.listing\_id), MAX(price\_per\_month) max\_price\_month

FROM reviews

JOIN prices

USING (listing\_id)

GROUP BY 1

HAVING COUNT(reviews.listing\_id)>=10

ORDER BY 3 DESC

LIMIT 5;

Bonus Question 4: Find the neighborhood(s) that have the highest variance in listing prices.

SELECT neighbourhood, ROUND(VARIANCE(price),2) variance\_

FROM prices

GROUP BY 1

HAVING VARIANCE(price) IS NOT NULL

ORDER BY 2 DESC;

Bonus Question 5: Calculate the average price\_per\_month for each neighborhood, taking into account only listings where the host has a minimum\_nights value that is higher than the average minimum\_nights value across all listings.

SELECT neighbourhood, ROUND(AVG(price\_per\_month)::DECIMAL,2) avg\_price\_month

FROM prices

JOIN reviews

USING (listing\_id)

WHERE reviews.minimum\_nights>(SELECT ROUND(AVG(reviews.minimum\_nights),2) FROM reviews)

GROUP BY 1

ORDER BY 2 DESC;