Airbnb Interview

Data Science Analytics

OKR Model

Objectives:

- Increase the number of bookings in Rio de Janeiro (RDJ).
- Attract new & retain existing customers in the RDJ market.
- Improve the guest-host matching process (speed + convenience).

Key Results:

- Total bookings increases over time.
- Cumulative customer base grows.
- Booking conversion increases over time.

Source: https://medium.com/productschool/how-to-measure-success-with-airbnbs-product-manager-1c10c562cf59

Key Metrics

- Conversion funnel:
  - # Interactions
  - # Replies
  - # Acceptances
  - # Bookings
- Reply Rate (%): # Replies / # Interactions
- Booking Rate (%): # Bookings / # Interactions (based on initial interaction date)
- Acceptance Rate (%): # Accepted Bookings / # Interactions
- Abandonment Rate (%): # Bookings *not* finalized, but were accepted
- Average time in-between:
  - Initial contact → first reply by host
  - Reply → acceptance
  - Acceptance → booking finalized
  - Start → Finish (complete funnel)
- Total # and % split of unique:
  - First-time customers
  - Repeat customers

Possible Key Metric Segmentations

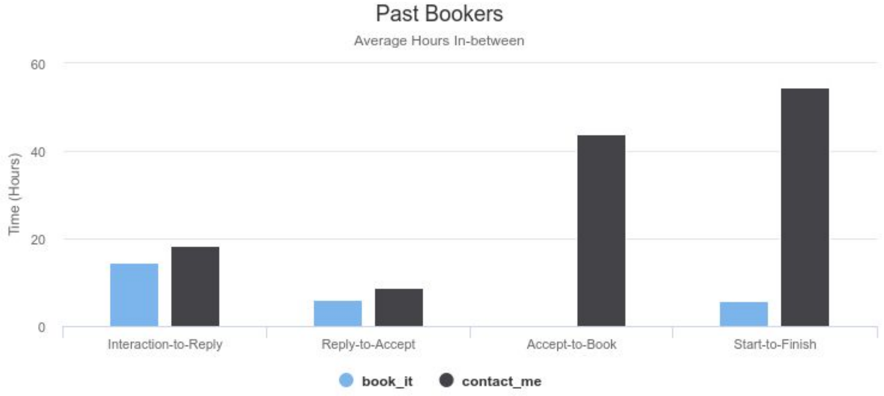
- Booking channel (Contact Me vs. Book It vs. Instant)
- New vs. existing customer
- Size of guest party
- Length of stay
- Accomodation type (e.g. entire apt/house, private/shared room, etc.)
- Neighborhood location (known vs. unknown)
- Guest origin (= or != host)
- “Completeness” of guest and host profiles (# words)
- Length of initial communication (# words)
- Total # messages exchanged between guest/host

Key Learning: “Contact Me” performs poorly

- For *“Contact Me”* channel (top row), there exists biggest drop-off @ last stage of booking funnel (light blue); that is, host accepts guest request, but guest does not convert (abandoned).
- This doesn’t happen with *“Book It”* (2nd row; above).

- Observe the abandonment rate (**red**; above), we see that for the *“Contact Me”* channel (top row), this metric is ~10x the same *“Book It”*.
- Drop-off between reply → acceptance (**purple** → **dk blue**) is about the same for both, suggesting issue with late-stage conversion for *“Contact Me”* feature flow.

“Contact Me” performs poorly (continued)



- For new & past bookers alike, the average time it takes from initial contact to 1st reply for the *"Contact Me"* channel (blue) is *much* slower than *"Book It"* (2x slower for new!)
- Terrible customer experience for guests when seeking travel accommodations (Why wait upwards of an entire day just for a response?)
- Those who used *“Contact Me”* are waiting a very long time (2+ days in some instances) after getting their booking accepted by host to actually confirm, if at all. Speculation: guests are indecisive or shopping around.
- Total time spent in booking flow is ~9-10x longer for *“Contact Me”* than *“Book It”*, which usually finishes within 12 hours (first message → booked).

Recommendation: Remove the *“Contact Me”* flow altogether in favor of *“Book It”* and *“Instant Book”* (where available).

No change to the guest experience; they’re still not charged until booking accepted by host. Removes friction, encourages guest to move forward within funnel with conviction.

KL: Active communication between guests/hosts is important

What if: Airbnb product manager says, *"We're keeping 'Contact Me' no matter what!!"*

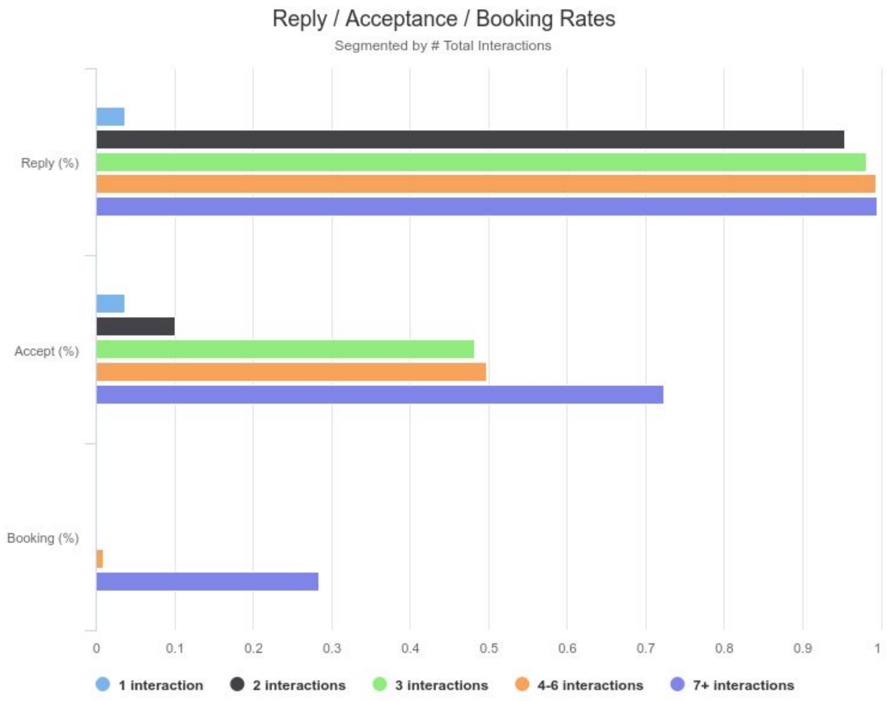
Goal then becomes: Improve conversion @ various stages of funnel, which in turn would drive add'l bookings & increased conversion.

Q: If we use the *# of guest-host interactions* a proxy for how "active" guests/hosts are in comms, does this have an effect on reply/accept/booking rates?

Intuitively, it follows that the more guests/hosts communicate with one another, the more comfortable they become and thus more likely to have an accepted/confirmed booking.

A: After segmenting data into ordinal, and testing statistical significance between groups (p-values < 2.2e-16; see appendix), we determine there is indeed positive relationship between Acceptance & Booking Rates, as the *# of interactions* increase. Also note, when there's only a single interaction (only guest sends message; no reply from host), accept/booking % is understandably low.

Recommendation: Require hosts to respond to initial message <u>and</u> encourage active communication between guests and hosts.



KL: Guests should try to make a good first impression

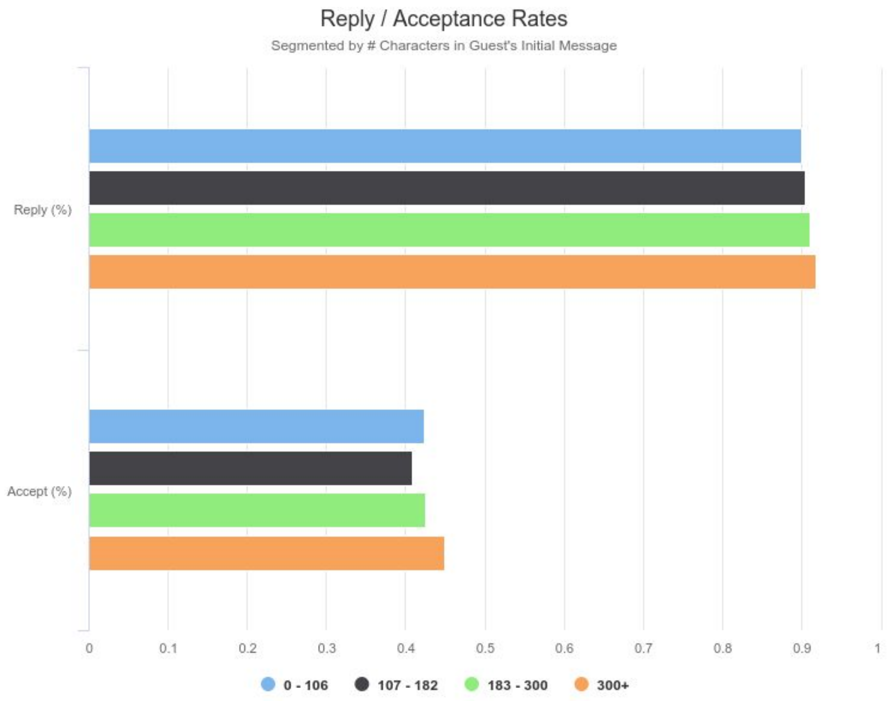
As we've shown, active communication between guest & host plays a significant role in whether a booking is ultimately accepted/booked.

Q: In previous, we saw that many initial messages are never replied to. What lever(s) can we pull to change that?

Let's examine the length of the guest's *first* message as a proxy for quality. Intuitively, a longer/better first message is more likely to garner a reply. Something short/simple such as "Hi, my name is Bob" probably isn't going to be as successful as a longer one that introduces yourself and explains why you're visiting Rio de Janeiro.

A: Just like with # comms, we create ordinal segmentation and compute multi-proportion test statistic. Even though the Reply % and Accept % increase by ~1-3% as first message length increases, the p-values are < 0.02 (see appendix) which suggests these differences are statistically significant.

Recommendation: Assuming we retain the *"Contact Me"* flow, we should require all potential guests to write an introductory message ~180 characters in length (50th percentile; not much longer than a Tweet) to not only increase the likelihood of a successful booking, but also improve the host's experience as they might want to know a little more about a potential stranger staying in their listing.



Additional Research & Experiments

- Research dynamics behind lead times
- Pricing methodologies & elasticity
    - What effect(s) does the daily booking rate have on demand/conversion?
    - Clustering of “similar” listings (e.g. “beach homes”, “penthouses with a view”, etc.)
    - Explore & test other pricing techniques (auction based, “Name Your Own Price”, “surge” pricing for seasonality/event-driven, etc.)
    - Bundled booking & pricing with “Experiences” product vertical
- Implement/test “Featured Listings”
    - Could be machine learning derived or human curated
- In markets where supply is limited or future expansion target but demand is known (say, based on search), get creative with creating supply through business development or real estate partnerships.

Appendix (R code)

```
rm(list=ls()); gc()  # Cleanup

library(tidyverse)
library(reshape2)
library(xts)
library(lubridate)
library(data.table)
library(highcharter)  # If you haven't used this charting package before, get the latest from CRAN: install.packages("highcharter")

options(tibble.width = Inf)  # Prints all columns in a tibble in the console

## Load data from .csv
contacts <- read_csv('contacts.csv')
listings <- read_csv('listings.csv')
users <- read_csv('users.csv')

## Join all 3 together into 1 giant master table:
contacts <- left_join(
  left_join(
    left_join(contacts, listings), users, by = c("id_guest_anon" = "id_user_anon")), users, by = c("id_host_anon" = "id_user_anon"), suffix = c("_guest", "_host")) %>%
  distinct() %>%  # Remove duplicates
  filter(guest_user_stage_first != '-unknown-' & total_reviews >= 0 & (ts_booking_at <= ds_checkout_first | is.na(ts_booking_at)))  # Assume rows where user's first stage is unknown
or review count is negative as junk data; also, you can't confirm a booking beyond the checkout date (that wouldn't make sense)
```

Appendix (R code)

```
## Create data.frames/tibbles:

# List of counts of interactions/replies/acceptances/bookings & proportions, aggregated by month, segmented by contact channel:
contacts_monthly <- sapply(X = c("contact_me", "book_it", "instant_book"), FUN = function(x) {
  foo <- contacts %>%
    filter(contact_channel_first == x) %>%
    mutate(ds_interaction_first = floor_date(ts_interaction_first, "month")) %>%  # group by month starting
    group_by(ds_interaction_first) %>%
    summarise(interactions = length(ts_interaction_first),
              replies = sum(!is.na(ts_reply_at_first)),
              acceptances = sum(!is.na(ts_accepted_at_first)),
              bookings = sum(!is.na(ts_booking_at))) %>%
    mutate(reply_rate = replies/interactions,
           accept_rate = acceptances/interactions,
           booking_rate = bookings/interactions,
           abandon_rate = accept_rate-booking_rate) %>%
    as.data.frame()
  return(xts(foo[-1], order.by = as.Date(foo[,1])))  # Convert to `xts` class time-series object
}, simplify = F)  # names(contacts_monthly)

# Time (hours) in-between funnel stages, segmented by new/existing customer & contact channel:
new_v_past_by_channel <- contacts %>%
  mutate(time_interaction_to_reply = difftime(ts_reply_at_first, ts_interaction_first, units = "hours"),
         time_reply_to_accept = difftime(ts_accepted_at_first, ts_reply_at_first, units = "hours"),
         time_accept_to_book = difftime(ts_booking_at, ts_accepted_at_first, units = "hours"),
         start_to_finish = difftime(ts_booking_at, ts_reply_at_first, units = "hours")) %>%  # Total time in the funnel
  group_by(guest_user_stage_first, contact_channel_first) %>%
  summarise(avg_time_interaction_to_reply = round(mean(time_interaction_to_reply, na.rm = T), 2),
            avg_time_reply_to_accept = round(mean(time_reply_to_accept, na.rm = T), 2),
            avg_time_accept_to_book = round(mean(time_accept_to_book, na.rm = T), 2),
            avg_start_to_finish = round(mean(start_to_finish, na.rm = T), 2))
```

Appendix (R code)

```
# List of flattened contacts with additional explanatory 0/1 variables for further manipulation:
contacts_flat <- sapply(X = c("contact_me", "book_it"), FUN = function(x) {contacts %>%
    mutate(length_of_stay_days = as.integer(difftime(ds_checkout_first, ds_checkin_first, units = "days")),
           replied = case_when(
             !is.na(ts_reply_at_first) ~ 1,
             TRUE ~ 0
           ),
           accepted = case_when(
             !is.na(ts_accepted_at_first) ~ 1,
             TRUE ~ 0
           ),
           booked = case_when(
             !is.na(ts_booking_at) ~ 1,
             TRUE ~ 0
           ),
           guest_host_same_country = case_when(
             country_guest == country_host ~ 1,
             TRUE ~ 0
           ),
           neighborhood_listed = case_when(
             listing_neighborhood == "-unknown-" ~ 0,
             TRUE ~ 1
           ),
           guest_profile_words = case_when(
             words_in_user_profile_guest <= 3 ~ "3 words or less",
             TRUE ~ "More than 3 words"
           ),
           host_profile_words = case_when(
             words_in_user_profile_host <= 3 ~ "3 words or less",
             TRUE ~ "More than 3 words"
           )) %>%
```

Appendix (R code)

```
    filter(contact_channel_first == x) %>%
    select(contact_channel_first,
           guest_user_stage_first,
           m_guests, m_interactions,
           m_first_message_length_in_characters,
           length_of_stay_days,
           room_type,
           total_reviews,
           listing_neighborhood,
           neighborhood_listed,
           guest_host_same_country,
           words_in_user_profile_guest,
           words_in_user_profile_host,
           guest_profile_words,
           host_profile_words,
           replied,
           accepted,
           booked)
}, simplify = F)
```

Appendix (R code)

```
## Charts

# Booking "funnel" for 3 channels:
highchart(type = "stock") %>%
  hc_title(text = "Contact Me vs. Book It vs. Instant") %>%
  hc_subtitle(text = "Monthly Interactions / Replies / Acceptances / Bookings") %>%
  hc_yAxis_multiples(
    create_yaxis(3, height = c(1, 1, 1), turnopposite = TRUE)
  ) %>%
  hc_add_series(contacts_monthly[["contact_me"]][,c("interactions")], yAxis = 0, name = "Contact Me - # Interactions", type = "column", color = "#e91e63") %>%
  hc_add_series(contacts_monthly[["contact_me"]][,c("replies")], yAxis = 0, name = "Contact Me - # Replies", type = "column", color = "#c2185b") %>%
  hc_add_series(contacts_monthly[["contact_me"]][,c("acceptances")], yAxis = 0, name = "Contact Me - # Acceptances", type = "column", color = "#3f51b5") %>%
  hc_add_series(contacts_monthly[["contact_me"]][,c("bookings")], yAxis = 0, name = "Contact Me - # Bookings", type = "column", color = "#2196f3") %>%

  hc_add_series(contacts_monthly[["book_it"]][,c("interactions")], yAxis = 1, name = "Book It - # Interactions", type = "column", color = "#e91e63") %>%
  hc_add_series(contacts_monthly[["book_it"]][,c("replies")], yAxis = 1, name = "Book It - # Replies", type = "column", color = "#c2185b") %>%
  hc_add_series(contacts_monthly[["book_it"]][,c("acceptances")], yAxis = 1, name = "Book It - # Acceptances", type = "column", color = "#3f51b5") %>%
  hc_add_series(contacts_monthly[["book_it"]][,c("bookings")], yAxis = 1, name = "Book It - # Bookings", type = "column", color = "#2196f3") %>%

  hc_add_series(contacts_monthly[["instant_book"]][,c("bookings")], yAxis = 2, name = "Instant - # Bookings", type = "column", color = "#2196f3")

# **Notes on above**: "Contact Me" channel has massive drop-off at last stage of the funnel, that is where a booking was accepted, but didn't end up getting booked (abandoned).
# Opportunity exists to increase bookings by reducing the drop-off at the Reply >> Acceptance & Acceptance >> Booking stages of the booking funnel.
```

Appendix (R code)

```
# Reply/Acceptance/Booking/Abandon rates for Contact Me vs. Book It:
highchart(type = "stock") %>%
  hc_title(text = "Contact Me vs. Book It") %>%
  hc_subtitle(text = "Monthly Reply / Acceptance / Booking / Abandonment Rates") %>%
  hc_yAxis_multiples(
    create_yaxis(2, height = c(1, 1), turnopposite = TRUE)
  ) %>%
  hc_add_series(round(contacts_monthly[["contact_me"]][,c("reply_rate")], 3), yAxis = 0, name = "Contact Me - Reply %", type = "column", color = "#673ab7") %>%
  hc_add_series(round(contacts_monthly[["contact_me"]][,c("accept_rate")], 3), yAxis = 0, name = "Contact Me - Accept %", type = "column", color = "#3f51b5") %>%
  hc_add_series(round(contacts_monthly[["contact_me"]][,c("booking_rate")], 3), yAxis = 0, name = "Contact Me - Booking %", type = "column", color = "#2196f3") %>%
  hc_add_series(round(contacts_monthly[["contact_me"]][,c("abandon_rate")], 3), yAxis = 0, name = "Contact Me - Abandon %", type = "column", color = "#f44336") %>%

  hc_add_series(round(contacts_monthly[["book_it"]][,c("reply_rate")], 3), yAxis = 1, name = "Book It - Reply %", type = "column", color = "#673ab7") %>%
  hc_add_series(round(contacts_monthly[["book_it"]][,c("accept_rate")], 3), yAxis = 1, name = "Book It - Accept %", type = "column", color = "#3f51b5") %>%
  hc_add_series(round(contacts_monthly[["book_it"]][,c("booking_rate")], 3), yAxis = 1, name = "Book It - Booking %", type = "column", color = "#2196f3") %>%
  hc_add_series(round(contacts_monthly[["book_it"]][,c("abandon_rate")], 3), yAxis = 1, name = "Book It - Abandon %", type = "column", color = "#f44336")

# **Notes on above**: As we saw in the first chart, the abandon rate for the "Contact Me" channel is super-high (10X the same metric for "Book It").
# Also, there's a near 40% drop-off between the reply >> acceptance stages, suggesting there's an opportunity to increase total bookings if we can close this gap.

# Average hours in-between:
reshape2::melt(as.data.frame(new_v_past_by_channel)) %>%
  filter(guest_user_stage_first == "new" & contact_channel_first != "instant_book") %>%  # Ignore Instant bookings since there is not any time in-between stages
  hchart("column", hcaes(x = "variable", y = "value", group = "contact_channel_first")) %>%
  hc_title(text = "New Bookers") %>%
  hc_subtitle(text = "Average Hours In-between") %>%
  hc_xAxis(title = NULL, categories = list("Interaction-to-Reply", "Reply-to-Accept", "Accept-to-Book", "Start-to-Finish")) %>%
  hc_yAxis(title = list(text = "Time (Hours)"))
```

Appendix (R code)

```
reshape2::melt(as.data.frame(new_v_past_by_channel)) %>%
  filter(guest_user_stage_first == "past_booker" & contact_channel_first != "instant_book") %>%
  hchart("column", hcaes(x = "variable", y = "value", group = "contact_channel_first")) %>%
  hc_title(text = "Past Bookers") %>%
  hc_subtitle(text = "Average Hours In-between") %>%
  hc_xAxis(title = NULL, categories = list("Interaction-to-Reply", "Reply-to-Accept", "Accept-to-Book", "Start-to-Finish")) %>%
  hc_yAxis(title = list(text = "Time (Hours)"))

# **Notes on above**: We see in the above that, for new/past bookers alike, the average time it takes from initial contact to 1st reply for the "Contact Me" channel is *MUCH* slower than "Book It".
# From a customer experience perspective, this is a clear negative as when searching for travel accommodations, one would intuitively prefer speed. For new customers especially, having to wait upwards of a day (on average) to get a response is simply unacceptable -- why not just book a hotel?
# Also, bookers using "Contact Me" are waiting a *VERY* long time after their bookings are accepted to actually book. Possible reason: users are shopping around for other listings or simply indecisive.
# Lastly, the total time spent in the funnel from first interaction to booking for "Contact Me" is 9-10x longer than "Book It" (which, on average, is within a half-day). This suggests a bad customer experience--nobody wants to spend *days* waiting around to book travel accommodations.

# Above we've seen that the "Contact Me" channel is clearly performing badly, and the obvious recommendation would be to nix it in favor of going "Book it" and "Instant Book" only.
# However, what if an Airbnb product manager said "Nah, we're keeping 'Contact Me' no matter what." What then?
# Well, the goal then would be to seek out ways to improve conversion @ various stages of the funnel, which in turn would [hopefully] drive additional bookings/increased conversion.

# Let's consider a few ideas at a couple different stages of the funnel.

# Q: If we use the # of interactions a proxy for how "active" guests/hosts are in comms, does this have an effect on reply/accept/booking rate?
# Intuitively, it follows that the more guests/hosts communicate with one another, the more comfortable they become and thus more likely to have an accepted/completed booking.
hchart(contacts_flat[["contact_me"]][["m_interactions"]])  # Quick histogram of interaction count; observe, skewed fat left tail.
```

Appendix (R code)

```
# Create ordinal segmentation based on # of interactions according to: quantile(contacts_flat[["contact_me"]][["m_interactions"]])
n_interactions <- contacts_flat[["contact_me"]] %>%
  mutate(n_interaction_group = case_when(
    m_interactions > 0 & m_interactions <= 1 ~ "1 interaction",
    m_interactions > 1 & m_interactions <= 2 ~ "2 interactions",
    m_interactions > 2 & m_interactions <= 3 ~ "3 interactions",
    m_interactions > 3 & m_interactions <= 6 ~ "4-6 interactions",
    m_interactions > 6 ~ "7+ interactions"
  )) %>%
  group_by(n_interaction_group) %>%
  summarise(n = n(),
            replies = sum(replied),
            accepts = sum(accepted),
            bookings = sum(booked),
            reply_rate = sum(replied)/n(),
            accept_rate = sum(accepted)/n(),
            booking_rate = sum(booked)/n())

reshape2::melt(n_interactions[,c("n_interaction_group", "reply_rate", "accept_rate", "booking_rate")]) %>%
  hchart("bar", hcaes(x = "variable", y = "value", group = "n_interaction_group")) %>%
  hc_title(text = "Reply / Acceptance / Booking Rates") %>%
  hc_subtitle(text = "Segmented by # Total Interactions") %>%
  hc_xAxis(title = NULL, categories = list("Reply (%)", "Accept (%)", "Booking (%)")) %>%
  hc_yAxis(title = "Percentage (%)", max = 1)

# Q: Is the above statistically significant?
prop.trend.test(n_interactions[["replies"]], n_interactions[["n"]])  # Use prop.trend.test() due to ordinal data
prop.trend.test(n_interactions[["accepts"]], n_interactions[["n"]])
prop.trend.test(n_interactions[["bookings"]], n_interactions[["n"]])
```

Appendix (R code)

```
# Based on the above chart & statistical tests (p-value's < 2.2e-16 for all 3), we see that there does appear to be a positive relationship between Acceptance & Booking Rates, as the
# of interactions increase.
# For those where there was only a *single* interaction (i.e. the potential guest sends the only message with no reply from the host), the Acceptance/Booking rates is understandbly
very low.
# The obvious recommendation here would be to *REQUIRE* hosts to respond to the initial guest message to improve the guest customer experience; in general, you would want to
encourage both guests/hosts to actively communicate with one another in the "Contact Me" feature flow, as that improves the likelihood of a successful booking.

# Q: Okay, so we see that active communication between guests/hosts plays a role in whether or not a booking gets accepted or ultimlately confirmed, but how can we drive that and
encourage hosts to actually respond?
# Let's take a look at the length of *FIRST* communication! Intuitively, one might expect a longer first message is more likely to get a reply from host.
# Something like "Hi, my name is Ray" probably isn't going to garner much of a response from a potential host, but a longer message that introduces yourself and why you're visiting
Rio de Janeiro probably would.

hchart(contacts_flat[["contact_me"]][["m_first_message_length_in_characters"]])  # Quick histogram of initial message lengths

# Again, let's create ordinal segmentation based on the above:
quantile(contacts_flat[["contact_me"]][["m_first_message_length_in_characters"]])  # Returns:
# 0%  25%  50%  75% 100%
# 0  107  183  301 1948

first_msg_length <- contacts_flat[["contact_me"]] %>%
  mutate(first_msg_nchar = case_when(
    m_first_message_length_in_characters >= 0 & m_first_message_length_in_characters < 107 ~ "0 - 106", # 25th percentile
    m_first_message_length_in_characters >= 107 & m_first_message_length_in_characters < 183 ~ "107 - 182",  # 50th percentile
    m_first_message_length_in_characters >= 183 & m_first_message_length_in_characters < 301 ~ "183 - 300",  # 75th percentile
    m_first_message_length_in_characters >= 301 ~ "300+"
  )) %>%
  group_by(first_msg_nchar) %>%
  summarise(n = n(),
    replies = sum(replied),
    accepts = sum(accepted),
    reply_rate = sum(replied)/n(),
    accept_rate = sum(accepts)/n())
```

Appendix (R code)

```
# Chart it:
reshape2::melt(first_msg_length[,c("first_msg_nchar", "reply_rate", "accept_rate")]) %>%
  hchart("bar", hcaes(x = "variable", y = "value", group = "first_msg_nchar")) %>%
  hc_title(text = "Reply / Acceptance Rates") %>%
  hc_subtitle(text = "Segmented by # Characters in Guest's Initial Message") %>%
  hc_xAxis(title = NULL, categories = list("Reply (%)","Accept (%)")) %>%
  hc_yAxis(title = "Percentage (%)", max = 1)

# Again, let's compute multi-proportion Chi-squared test statistic:
prop.trend.test(first_msg_length[["replies"]], first_msg_length[["n"]])  # p-value = 0.01428
prop.trend.test(first_msg_length[["accepts"]], first_msg_length[["n"]])  # p-value = 0.008349
prop.test(c(sum(first_msg_length[["replies"]][1:2]), sum(first_msg_length[["replies"]][3:4])), c(sum(first_msg_length[["n"]][1:2]), sum(first_msg_length[["n"]][3:4]))) # Also do it for two groups @ the 50th percentile
# returns: p-value = 0.01714
prop.test(c(sum(first_msg_length[["accepts"]][1:2]), sum(first_msg_length[["accepts"]][3:4])), c(sum(first_msg_length[["n"]][1:2]), sum(first_msg_length[["n"]][3:4]))) # Also do it for two groups @ the 50th percentile
# returns: p-value = 0.01621

# Based on the above, we reject the null hypothesis with 95% confidence; that is, there *is* a statistically significant relationship between the length of a guest's initial message and the likelihood of a reply, and ultimately accepted booking under the "Contact Me" feature flow.
# Even though the difference *seems* small (~ 1-2% in reply rate; ~2-3% in acceptance), the p-value's being < 0.02 confirm this is something we shouldn't overlook.
# Recommendation: If we continue with the "Contact Me" flow, require potential guests to write an introductory message longer than approx. 180 characters -- which to be fair, isn't much longer than a Tweet.
# This makes for a better host experience as one might want to know a little bit more about who is staying in their listing before accepting their request.
```

Appendix (R code)

```
##################################################################################
# Cumulative count of new/past bookers, over time: (**unused in presentation**)
new_vs_past <- contacts %>%
  mutate(ds_interaction_first = floor_date(ts_interaction_first, "day")) %>%
  group_by(ds_interaction_first) %>%
  summarise(n_guest = n_distinct(id_guest_anon),
    n_new = data.table::uniqueN(id_guest_anon[guest_user_stage_first == "new" ]),
    n_past = data.table::uniqueN(id_guest_anon[guest_user_stage_first == "past_booker"])) %>%
  mutate(pct_new = n_new/n_guest, pct_past = n_past/n_guest, cum_n_new = cumsum(n_new), cum_n_past = cumsum(n_past), cum_pct_new = cumsum(n_new)/cumsum(n_guest), cum_pct_past =
cumsum(n_past)/cumsum(n_guest)) %>%
  as.data.frame()

new_vs_past.xts <- xts(new_vs_past[-1], order.by = as.Date(new_vs_past[,1]))

highchart(type = "stock") %>%
  hc_title(text = "Daily New vs. Past Bookers") %>%
  hc_yAxis_multiples(
    create_yaxis(3, height = c(1, 1, 1), turnopposite = TRUE)
  ) %>%1
hc_add_series(new_vs_past.xts[,c("n_new")], yAxis = 0, name = "# New", type = "line", color = "#e91e63") %>%
  hc_add_series(new_vs_past.xts[,c("n_past")], yAxis = 0, name = "# Past", type = "line", color = "#3f51b5") %>%

  hc_add_series(new_vs_past.xts[,c("cum_n_new")], yAxis = 1, name = "# New (Cumulative)", type = "line", color = "#e91e63") %>%
  hc_add_series(new_vs_past.xts[,c("cum_n_past")], yAxis = 1, name = "# Past (Cumulative)", type = "line", color = "#3f51b5") %>%

  hc_add_series(new_vs_past.xts[,c("cum_pct_new")], yAxis = 2, name = "% New (Cumulative)", type = "line", color = "#e91e63") %>%
  hc_add_series(new_vs_past.xts[,c("cum_pct_past")], yAxis = 2, name = "% Past (Cumulative)", type = "line", color = "#3f51b5")

# **Notes on above**: The cumulative # of new & past users is growing linearly since 1/1/16 (Q: did Airbnb guys launch RDJ in Jan'16? There's a spike in the beginning); the former is
growing at a faster rate, and the latter isn't plateauing--these both suggest a healthy market.
# There also aren't any abnormal spikes in the data as well (good thing) and the cumulative split % between new/past is stabilizing, but not converging. Convergence would probably be
a bad sign as you want continuous growth; cum % existing customers shouldn't overtake news ones would mean stalling new customer acquisition.
```

Appendix (R code)

```
# Q: Does the neighborhood, known or unknown, make a guest more likely to book?  (**unused in presentation**)
contacts_flat[["contact_me"]] %>%
  group_by(neighborhood_listed) %>%
  summarise(n = n(),
    accepts = sum(accepted),
    bookings = sum(booked),
    abandons = sum(accepted)-sum(booked),
    accept_rate = sum(accepted)/n(),
    booking_rate = sum(booked)/n(),
    abandon_rate = (sum(accepted)-sum(booked))/n()
  ) %>%
  arrange(desc(n))

# Quick 2 sample Z-test on booking rates where neighborhood is listed (or not)
prop.test(x = c(524, 377), n = c(6850,  5919), alternative = "two.sided")  # p-value of 0.005394, thus we reject H0 with 95% confidence; that is, there is statistical significance in booking rate when neighborhood is/isn't listed

# Recommend: improving geo-tagging so that listing neighborhood is mandatory.
```