TAKE-HOME CHALLENGE: Data Science - Analytics

Airbnb is a two sided marketplace which matches guests to hosts. The booking flow at Airbnb is as follows: a guest finds an available room (listing) that they like, and then they contact the host. Once the guest finds a listing they are interested in, there are three ways to send the host an inquiry: 'contact_me', 'book_it', or 'instant_book' (detailed at the bottom of this document). Upon receiving the inquiry, the host can then decide whether or not to accept the request (for 'contact_me' and 'book_it' methods -- `instant_book` is auto-accepted). One of our goals at Airbnb is to increase bookings on our platform.

Prompt:

You are the first data scientist to join a cross-functional Product and Operations team working to grow bookings in Rio de Janeiro. The team asks you for help with the following:

1. What key metrics would you propose to monitor over time the success of the team's efforts in improving the guest host matching process and why? Clearly define your metric(s) and explain how each is computed.
2. What areas should we invest in to increase the number of successful bookings in Rio de Janeiro? What segments are doing well and what could be improved? Propose 2-3 specific recommendations (business initiatives and product changes) that could address these opportunities. Demonstrate rationale behind each recommendation AND prioritize your recommendations in order of their estimated impact.
3. There is also interest from executives at Airbnb about the work you are doing, and a desire to understand the broader framing of the challenge of matching supply and demand, thinking beyond the data provided. What other research, experiments, or approaches could help the company get more clarity on the problem?

Your assignment: Summarize your recommendations in response to the questions above in a 5-8 slide presentation intended for the Head of Product and VP of Operations (who is not technical). Include an organized appendix sharing the details of your work conducted for the Rio team, that would be useful for the data team to understand your work.

Instructions:

1) Create a PDF of your presentation.
2) Append all code you use to analyze results to the above PDF, including code used for initial data exploration. We typically see data processed in SQL/R/Python and a presentation with results made in Keynote/Google slides/Powerpoint. But you are welcome to use any software you feel comfortable with. If you use Excel, please document the operations used to process the data, and append your spreadsheet.
3) Please do NOT include your name or email address on this PDF.
4) You will have 48 hours to complete the assignment.

Grading:

Your assignment will be judged according to:

1) The analytical approach and clarity of your graphs, tables, visualizations,
2) The data decisions you made and reproducibility of the analysis,
3) Strength of recommendations, prioritizations, and rationale behind those,
4) The narrative of your presentation and ability to effectively communicate to non-technical executives,
5) How well you followed the directions.

Data Provided:

Contacts - contains a row for every time that an user makes an inquiry for a stay at a listing in Rio de Janeiro.

- id_guest_anon - id of the guest making the inquiry.
- id_host_anon - id of the host of the listing to which the inquiry is made.
- id_listing_anon - id of the listing to which the inquiry is made.
- ts_interaction_first - UTC timestamp of the moment the inquiry is made.
- ts_reply_at_first - UTC timestamp of the moment the host replies to the inquiry, if so.
- ts_accepted_at_first - UTC timestamp of the moment the host accepts the inquiry, if so.
- ts_booking_at - UTC timestamp of the moment the booking is made, if so.
- ds_checkin_first - Date stamp of the check-in date of the inquiry.
- ds_checkout_first - Date stamp of the check-out date of the inquiry.
- m_guests - The number of guests the inquiry is for.
- m_interactions - The total number of messages sent by both the guest and host.
- m_first_message_length_in_characters - Number of characters in the first message sent by the guest, if a message was sent
- contact_channel_first - The contact channel through which the inquiry was made. One of {contact_me, book_it, instant_book}. *See bottom of page for more detail*
- guest_user_stage_first - Indicates whether the user has made a booking before sending the inquiry (“past booker”). If the user has not booked before, then the user is a new user.

Listings - contains data for every listing in the market

- id_listing_anon - anonymized id of the listing
- room_type - indicates whether the room is an entire home, private room, or shared room
- listing_neighborhood - the neighborhood of the listing
- total_reviews - the total number of reviews of the listing (at the time the data was pulled).

Users - contains data for every user

- id_user_anon - anonymized id of user
- words_in_user_profile - the number of words in the “about me” section of the user’s Airbnb profile (at the time of contact)
- country - origin country of the user

Further Information:

There are three ways to book a listing on Airbnb:

1) **contact_me** - The guests writes a message to the host to inquire about the listing. The host has the option to 1) pre-approve the guest to book their place, or 2) they can reject, or 3) they can write a free text message with no explicit acceptance or rejection. If the host pre-approves, the guest can then go ahead and click to make the booking (but is not obligated to).

2) **book_it** - The guest puts money down to book the place directly, but the host has to accept the reservation request. If the host accepts, the booking happens automatically. If you have used Airbnb before, this shows up as a button labeled “Request to book”.
3) **instant_book** - The guest books the listing directly, without any need for the host to accept or reject actively (it is auto-accepted by the host). This shows up as a button labeled “Book”.

Note: A host can opt-in to the `instant_book` feature. If a host does so, a guest can use the `contact_me` or `instant_book` channels for booking that particular listing, but cannot use the `book_it` functionality. Alternatively, if a host does not opt in, a guest can use the `contact_me` or `book_it` channels only. We suggest that you browse the Airbnb website and look at listings to see the different ways that you can message a host.

Rio de Janeiro:

Booking Opportunities

Product and Operations Team

Singapore 2018

Summary

To improve the performance, we should focus on:

- Helping New Brazilian Users get bookings *
- Lowering the usage of the “Contact Me” method by Brazilians *

Measure our initiatives with:

- Three New User Ratios
- Three “Contact Me” Ratios

We should also:

- Design a series of controlled experiments to test different initiatives that target New Brazilian users.
- Better understand the use of the “Contact Me” option. Mine the “Contact Me” messages for common themes and also analyze the browsing behavior.

airbnb

* Initial modeling showed that these two factors strongly influence the probability of a successful booking. Please check the Modeling section of the appendix for more information

Help New Users Get the Bookings

- Only 27% of New Brazilian Users booking attempts are successful (compare to 42% for New Foreign Users)
- 69% of Brazilian booking attempts come from New Users which represents a huge population that is not doing well **
- New Brazilian Users have less complete profiles and use less words when contacting hosts. Fixing these issues could help with the issue ***

	Conversion Rate *	
	Foreigners	**Brazilians**
New Users	42%	**27%**
Existing Users	59%	**40%**

* Booking Count / First Contact Count

airbnb

** Please check the New Users Analysis in Appendix for an extended explanation
*** Please check the Other Research Points in Appendix for more info

New Brazilian Users Bookings Monitoring

- To monitor our progress, we propose three metrics for New Brazilian Users. Metrics should be tracked by week.
  - **All Channel New Brazilian Users Conversion Rate:**
    - Unique Booking Count / User Initiated First Contact Count
    - This metric measures an overall conversion rate for New Brazilian Users
  - **“Contact Me” Channel, New Brazilian Users Conversion Rate:**
    - Unique Booking Count / User Initiated First Contact Count
    - “Contact Me” channel only
  - **New Brazilian Users Bookings Distribution By Contact Channel:**
    - Unique Booking Percentage Distribution by contact channel
    - Helps us see if nudging users to use different channels is effective over time

Design experiments to help New Brazilian Users

Create a series of controlled experiments to test different initiatives:

- **Search**:
  - Increase search ranking of hosts who don't mind new users (have a high conversion rate with new users) or have many reviews (this feature was associated with higher chances of conversion for new users).*
  - Add a special badge to search result listings of hosts that are New User friendly.
- **New Brazilian Users**
  - Message new users the benefits of having a complete profile.
  - Require new users to input personal information at the registration phase. **
  - Suggest a minimum number of words for the initial "Contact me" message
- **Provide a financial incentive to hosts for accepting New Users.**

airbnb

* Please check New Users Analysis (Part 3) in the Appendix for more information

** We have to be careful here so we do not scare off New Users at registration (monitor the registration drop off rates)

Lower the “Contact Me” bookings rates

- Only 6% of Brazilian “Contact Me” attempts end up with a booking.
- 58% of Brazilian booking attempts start with a “Contact Me” (35% for foreigners). This share goes up to 75% for New Brazilian Users.**
- We should focus on shifting Brazilian users to “Book It” or “Instant Book” options.

	Conversion Rate *	
	Foreigners	**Brazilians**
Contact Me	9%	**6%**
Book It	51%	**42%**
Instant Book	100%	100%

* Booking Count / First Contact Count

** Please check the “Contact Me” Analysis in the Appendix for more information

“Contact Me” Bookings Monitoring

- To monitor our progress, we propose three metrics. These metrics should be tracked by week.
  - **Bookings By Contact Channel:**
    - Unique Booking Percentage Distribution by contact channel.
  - **All Listings Distributions By Contact Channel:**
    - All Unique Listings Percentage Distributions by contact channel.
    - This measures if we decrease the number of listed properties that just offer the “Contact Me” booking option.
  - **Booked Listings Distributions By Contact Channel:**
    - Booked Unique Listings Percentage Distributions by contact channel.
    - Measures if more unique properties are being booked by other channels. *

airbnb

* It is one thing to have listings with other booking options and something else for Users to use it. A host might have different booking options, but for whatever reason people might still choose to go with “Contact Me”

Better Understand “Contact Me”

We need to understand why Brazilians prefer the “Contact Me” option.

1. **Explore the browsing behavior of Brazilian users**. There is a possibility that the Brazilian users utilize the “Contact Me” option as the listings they are interested in only offer this way of booking. If that is the case, then we should work with hosts to better understand why they do not want to have other booking options for their listings and fix it.
2. **Data mine communication between the host and the user**. Try to find common topic or questions in those messages. This information could be provided to the hosts so they can include it with the property listing or with enough data, we can design an automated reply system.

Appendix

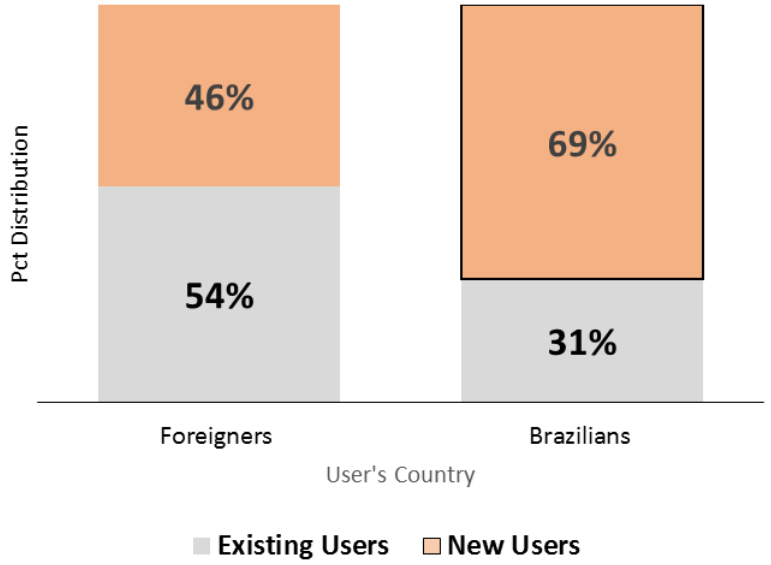
New Users Analysis (Part 1)

69% of booking contacts came from Brazilian New Users (compared to 46%)

(We recognize that there is some self selection bias due to the fact that we look at Rio and Brazilians, therefore, we would need to compare these numbers to other countries)

Brazilians have lower conversion rates (-15% New User, -19% Existing Users)

We want to focus on New Users as they represent a larger user base and gains in this group will potentially translate into a larger group of future existing users



	Conversion Rate *	
	Foreigners	**Brazilians**
New Users	42%	**27%**
Existing Users	59%	**40%**

* Booking Count / First Contact Count

New Users Analysis (Part 2)

- Successful bookings had on average more words in their first message (between 10% and 30%)
- Successful bookings for new users had more words (15% and 30%) than for existing users (10% and 19%)

	Word Count Of The First Message	
	Foreigners	**Brazilians**
Failed Booking		
New User	268	206
Existing User	267	192
Successful Booking		
New User	307	267
Existing User	294	229
Successful - Failed Booking		
New User	**40**	**61**
Existing User	**27**	**37**
New User (% Dif)	**15%**	**30%**
Existing User (% Dif)	**10%**	**19%**

* “Contact Me” Only

New Users Analysis (Part 3)

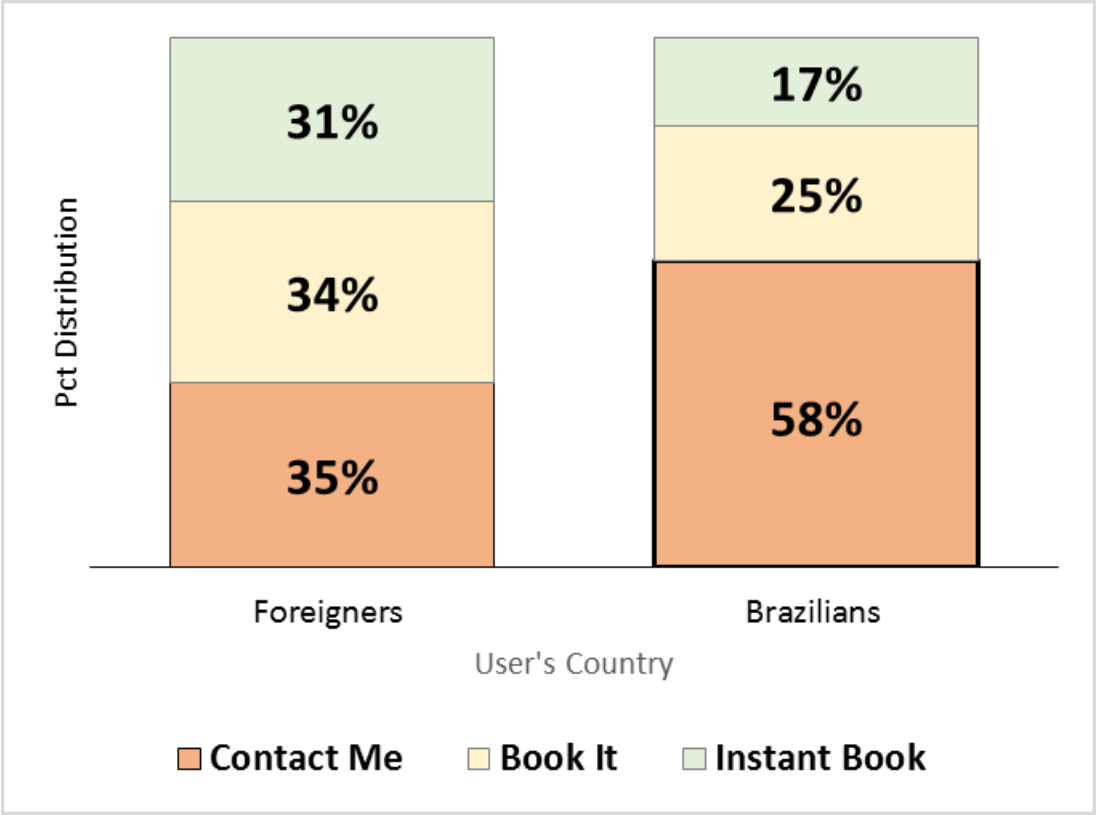
- Brazilians have shorter first message to the host (-61.9 and -40.2 characters)
- Also, they have much shorter user profiles (-3.2 and -7.0 words)
- Brazilians book listings with more reviews (3.3)
- It can be assumed that incomplete profiles and less information in messages might be a reason for lower conversion rate (trust?)

"Contact Me" and New Users Only

	Avg. Len. of First Message (characters)		Avg. Len. Of User Profile (words)		Avg. Num. Of Listing Reviews	
	No Booking	Booked	No Booking	Booked	No Booking	Booked
Brazilians	206.0	267.3	3.8	2.7	8.7	17.6
Foreigners	267.9	307.4	7.0	9.7	9.0	14.3
Difference	**-61.9**	**-40.2**	**-3.2**	**-7.0**	**-0.3**	**3.3**

“Contact Me” Analysis (Part 1)

- 58% of Brazilians initiate the contact with a host via “Contact Me” option
- A big question is why Brazilians do not use the Instant Book Option?
- Is it the trust issue on the host and user’s sides?



“Contact Me” Analysis (Part 2)

- 75% of “Contact Me” messages come from New Users
- Why do they prefer this type of booking?
- What information do they seek?
- How can we improve it?

"Contact Me"

	Foreigners		Brazilians	
	Contact Share	**Booking Share**	**Contact Share**	**Booking Share**
New User	59%	50%	**75%**	**69%**
Existing User	41%	50%	**25%**	**31%**

Modeling : Intro

- In order to better understand what impacts the bookings, a linear logistic model was built
- It is important to understands that the models are limited by the availability of data and that some of the observed results might be due to unaccounted variables
  - The country of origin is a good example of a potentially misleading variable. Some models showed that being from Brazil decreases the probability of booking. However, there might be other explanations to this fact. Maybe it is due to unobserved user characteristics or shortcomings in user profile rather than the fact that the user is from Brazil
  - However, we should also look into the potential fact that maybe our hosts do not want bookings from Brazilian guests. If that is the case, the we should investigate the origins of this situation and act accordingly

Modeling 1: Only Brazilian Bookings

- "Contact Me" and being a New User have the strongest negative impact on the probability of getting a successful booking(-3.73, -0.28)

0	contact_me_flag	[-3.727980057086825]
1	new_user_flag	[-0.2802596825210466]
2	location_unknown_flag	[0.007394300562890962]
3	m_interactions	[0.16178508753142082]
4	m_first_message_length_in_characters	[-0.00027934023201824586]
5	m_guests	[-0.0677917224496325]
6	words_in_user_profile	[-0.0003239430472334989]
7	room_apt_flag	[0.04317297035229868]
8	room_private_flag	[0.15169835593109418]
9	total_reviews_pos	[0.016152795701454573]
10	interact_2_check_in_days	[-0.004778881598572779]
11	stay_duration_days	[-0.04807451054195284]

Model Details: Logistic regression on Only Brazilian Rio users, predicting the probability of being a Brazilian. 87% F1 and Recall

Modeling 2: What Defines a Brazilian User?

- "Contact Me" , being a New User, and booking a place with an unknown location have the strongest positive relationship with being a Brazilian User (0.85, 0.81, and 0.1 respectively)
- Booking a place had the strongest negative relationship with being a Brazilian (-0.34)

0	book_flag	[-0.34201324930188415]
1	contact_me_flag	[0.8519732109502396]
2	new_user_flag	[0.8064358076108182]
3	location_unknown_flag	[0.10082447292492748]
4	m_interactions	[-0.007633085505490978]
5	m_first_message_length_in_characters	[-0.0032097730588358606]
6	m_guests	[0.057998247521943554]
7	words_in_user_profile	[-0.004507536320170442]
8	room_apt_flag	[0.19906801050853667]
9	room_private_flag	[0.1344828056317339]
10	total_reviews_pos	[-0.004954566784209933]
11	interact_2_check_in_days	[-0.002051140877476678]
12	stay_duration_days	[-0.013971190160498262]

Model Details: Logistic regression on All Rio users, predicting the probability of being a Brazilian. 69% F1 and Recall

Modeling 3: What Influences Booking (All Users)?

- “Contact Me” , being from Brazil, and being a New User have the strongest negative impact on the probability of getting a successful booking(-3.7, -0.4 and -0.2 respectively)

0	contact_me_flag	[-3.6916352088130915]
1	new_user_flag	[-0.17481105582023848]
2	br_country_flag	[-0.4091219692434835]
3	location_unknown_flag	[-0.027358793098680823]
4	m_interactions	[0.2123524918073225]
5	m_first_message_length_in_characters	[-0.0005528026690888459]
6	m_guests	[-0.048911079777550534]
7	words_in_user_profile	[0.00021131491862111 71]
8	room_apt_flag	[0.042928435199020246]
9	room_private_flag	[0.12936798408461705]
10	total_reviews_pos	[0.018582926382815314]
11	interact_2_check_in_days	[-0.004755524170973826]

Model Details: Logistic regression on All users, predicting the probability of a successful booking. 87% F1 and Recall

Other Research Points

- One theory we explored was that Brazilians contact more hosts for the same travel period. Did not find a strong support for this claim
- “Contact Me” was preferred by Brazilians with all types of rooms
- On average, Brazilians and foreigners that used “Contact Me” had a slightly more guests
- Overall, foreigners had:
  - More interactions (ask more questions?)
  - Longer messages
  - Longer user profiles
  - Considered places with more reviews

In [46]:
```
import pandas as pd
import numpy as np
```

Upload Data

In [47]:
```
contacts = pd.read_csv('contacts.csv')
```

In [48]:
```
contacts.describe()
```

Out[48]:

	m_guests	m_interactions	m_first_message_length_in_characters
count	27886.000000	27887.000000	27887.000000
mean	2.782041	8.255818	193.712626
std	1.863797	10.035784	183.172132
min	0.000000	1.000000	0.000000
25%	2.000000	3.000000	66.000000
50%	2.000000	5.000000	154.000000
75%	4.000000	10.000000	269.000000
max	16.000000	410.000000	2341.000000

In [294]:
```
contacts.head()
```

0	da8656a1-51af-4f38-b1c4-94be1f585157	5426897d-960d-4013-9e38-606ae746793c	a408a8b2-0d44-4513-a611-3736d0409bb2	2016-04-21 02:55:53	2016-04-21 03:15:00.0	2
1	8590d6f1-8bc9-4e8b-bdfb-de78f69fcf37	f30417c5-6df4-45ac-bfc2-6ad1cce398ab	e387c705-0aeb-464c-9375-ece63be6f006	2016-02-16 22:14:01	2016-02-16 23:37:36.0	
2	ebcd83ba-bda1-47eb-9680-2dd04ccf3988	13cbf50a-3272-45d4-9866-a06b6ea1b99a	d1eb1960-938f-4305-a353-51e224414dd4	2016-01-27 23:33:38	2016-01-28 02:12:47.0	
3	b0af8848-fe2a-4ef1-991e-26ab3066feb3	01614601-d5a4-4776-ab9b-c10d3b865bf0	855f6779-346c-45fc-a64b-ea133250ca7a	2016-05-05 14:42:52	2016-05-05 15:17:40.0	2

In [50]:
```
listings = pd.read_csv('listings.csv')
```

In [51]:
```
# -34 in the number of reviews (that is an error)
listings.describe()
```

Out[51]:

	total_reviews
count	13038.000000
mean	7.004525
std	17.059747
min	-34.000000
25%	0.000000
50%	1.000000
75%	5.000000
max	268.000000

In [295]:
```
listings.head()
```

Out[295]:

	id_listing_anon	room_type	listing_neighborhood	total_reviews
0	71582793-e5f8-46d7-afdf-7a31d2341c79	Private room	-unknown-	0.0
1	a1a3f728-e21f-4432-96aa-361d28e2b319	Entire home/apt	Copacabana	0.0
2	353a68be-ecf9-4b7b-9533-c882dc2f0760	Entire home/apt	Barra da Tijuca	3.0
3	b9ae1908-0486-40ac-bbcf-bd63ffebfe63	Entire home/apt	Lapa	4.0
4	fa0290ef-7881-4482-8981-8ebb1ce5dbfd	Entire home/apt	-unknown-	0.0

In [53]:
```
users = pd.read_csv('users.csv')
```

In [54]:
```
users.describe()
```

Out[54]:

	words_in_user_profile
count	31525.000000
mean	16.222141
std	44.391491
min	0.000000
25%	0.000000
50%	0.000000
75%	12.000000
max	1460.000000

In [296]:
```
users.head()
```

Out[296]:

	id_user_anon	country	words_in_user_profile
0	1d16a001-31a2-494c-a101-1f7b308adc62	FR	0
1	42607e0a-86c0-472e-b633-9e192114e93c	AR	0
2	25f85eb5-a700-44e1-b142-4c076222198d	BR	0
3	55abeba0-18ef-4c58-80f4-3c278b706aca	BR	1
4	5d62d35a-7d6d-45dd-ae9b-a5d2f82a7d7b	BR	98

Clean Transform Data

Check For Missing Data

Contacts

In [56]:
```
total = contacts.isnull().sum().sort_values(ascending = False)
percent = (contacts.isnull().sum()/contacts.isnull().count()*100).sort_values(
missing_contacts  = pd.concat([total, percent], axis=1, keys=['Total', 'Percen
missing_contacts.head(20)
```

Out[56]:

	Total	Percent
ts_booking_at	16300	58.450174
ts_accepted_at_first	11472	41.137448
ts_reply_at_first	2032	7.286549
m_guests	1	0.003586
guest_user_stage_first	0	0.000000
contact_channel_first	0	0.000000
m_first_message_length_in_characters	0	0.000000
m_interactions	0	0.000000
ds_checkout_first	0	0.000000
ds_checkin_first	0	0.000000
ts_interaction_first	0	0.000000
id_listing_anon	0	0.000000
id_host_anon	0	0.000000
id_guest_anon	0	0.000000

In [57]:
```
contacts[contacts['m_guests'].isnull()]
```

Out[57]:

	id_guest_anon	id_host_anon	id_listing_anon	ts_interaction_first	ts_reply_at_first	ts_ac
24738	abc1b137-28b0-4337-a93c-ed77c16f49d8	1a260779-97eb-4c3f-a020-c09a094033a4	f4ada617-3298-4c78-97ee-104edd981e09	2016-01-11 12:08:54	2016-01-12 17:48:59.0	

In [58]:
```
# Check if the guest had any recent bookings to estimate the m_guests (nope)
contacts[contacts['id_guest_anon'] == 'abc1b137-28b0-4337-a93c-ed77c16f49d8']
```

Out[58]:

	id_guest_anon	id_host_anon	id_listing_anon	ts_interaction_first	ts_reply_at_first	ts_ac
24738	abc1b137-28b0-4337-a93c-ed77c16f49d8	1a260779-97eb-4c3f-a020-c09a094033a4	f4ada617-3298-4c78-97ee-104edd981e09	2016-01-11 12:08:54	2016-01-12 17:48:59.0	

In [59]:
```
# The rest of the 'NULLs' just did not have a reply or accept
```

Listings

In [60]:
```
# Clean
total = listings.isnull().sum().sort_values(ascending = False)
percent = (listings.isnull().sum()/listings.isnull().count()*100).sort_values(
missing_listings = pd.concat([total, percent], axis=1, keys=['Total', 'Percent
missing_listings.head(20)
```

Out[60]:

	Total	Percent
total_reviews	0	0.0
listing_neighborhood	0	0.0
room_type	0	0.0
id_listing_anon	0	0.0

Users

In [61]:
```
total = users.isnull().sum().sort_values(ascending = False)
percent = (users.isnull().sum()/users.isnull().count()*100).sort_values(ascend
missing_users = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])
missing_users.head(20)
```

Out[61]:

	Total	Percent
country	1	0.003172
words_in_user_profile	0	0.000000
id_user_anon	0	0.000000

In [62]:
```
users[users['country'].isnull()]
```

Out[62]:

	id_user_anon	country	words_in_user_profile
18866	e0e03982-682e-4a58-a3f7-9f393e4875d6	NaN	0

In [63]:
```
# Check if this user is in the data set (that is the only user with no country
users[users['id_user_anon'] == 'e0e03982-682e-4a58-a3f7-9f393e4875d6']
```

Out[63]:

	id_user_anon	country	words_in_user_profile
18866	e0e03982-682e-4a58-a3f7-9f393e4875d6	NaN	0

Merge all 3 files together

In [64]:
```
contacts_users = pd.merge(contacts, users, left_on='id_guest_anon', right_on='
```

In [110]:
```
# contacts_users.head()
```

In [66]:
```
contacts_all = pd.merge(contacts_users, listings, left_on='id_listing_anon', r
```

In [297]:
```
contacts_all.head()
```

Out[297]:

	id_guest_anon	id_host_anon	id_listing_anon	ts_interaction_first	ts_reply_at_first	ts_accept
0	da8656a1-51af-4f38-b1c4-94be1f585157	5426897d-960d-4013-9e38-606ae746793c	a408a8b2-0d44-4513-a611-3736d0409bb2	2016-04-21 02:55:53	2016-04-21 03:15:00.0	
1	8590d6f1-8bc9-4e8b-bdfb-de78f69fcf37	f30417c5-6df4-45ac-bfc2-6ad1cce398ab	e387c705-0aeb-464c-9375-ece63be6f006	2016-02-16 22:14:01	2016-02-16 23:37:36.0	
2	ebcd83ba-bda1-47eb-9680-2dd04ccf3988	13cbf50a-3272-45d4-9866-a06b6ea1b99a	d1eb1960-938f-4305-a353-51e224414dd4	2016-01-27 23:33:38	2016-01-28 02:12:47.0	
3	b0af8848-fe2a-4ef1-991e-26ab3066feb3	01614601-d5a4-4776-ab9b-c10d3b865bf0	855f6779-346c-45fc-a64b-ea133250ca7a	2016-05-05 14:42:52	2016-05-05 15:17:40.0	
4	5ddbbcc3-ac1a-4d8a-b6f8-0f675b76d1b2	f2fed6f3-4c5c-453d-9e64-37c62b8bd06d	f2928a59-c5e7-42b2-9c37-240a40f38dc0	2016-06-23 03:09:25.0	2016-06-23 03:09:26.0	

5 rows × 27 columns

In [68]:
```
contacts_all.describe()
```

Out[68]:

	m_guests	m_interactions	m_first_message_length_in_characters	words_in_user_prof
count	27978.000000	27979.000000	27979.000000	27979.0000
mean	2.782436	8.247686	193.743879	10.2024
std	1.866441	10.025091	183.169421	30.8467
min	0.000000	1.000000	0.000000	0.0000
25%	2.000000	3.000000	66.000000	0.0000
50%	2.000000	5.000000	154.000000	0.0000
75%	4.000000	10.000000	269.000000	1.0000
max	16.000000	410.000000	2341.000000	658.0000

In [298]:
```
# Lets handle that -34 reviews
contacts_all[contacts_all['total_reviews'] < 0].head()
```

Out[298]:

	id_guest_anon	id_host_anon	id_listing_anon	ts_interaction_first	ts_reply_at_first	ts_acc
1282	8092c7ee-1bdc-4587-8aab-71268b6c6d1d	6c319423-9406-407d-a662-973862ee1e7c	46ed561a-497c-4de8-b120-47f42ffc9159	2016-03-20 16:56:24	2016-03-21 15:41:45.0	
1635	b804353a-1862-4eb4-b426-4b7ecd941859	9e85e799-4285-4848-8c2a-0872858aba75	b2a45ced-8c99-41b0-95a1-f23b993cdc87	2016-04-19 17:14:09	2016-04-19 18:23:02.0	
1851	03384592-dc7b-43f6-aee0-7ffbf92f19e4	1fb3add2-d916-48c1-b114-393aeef46506	e6d2e0c5-b17a-45f1-b99f-7d086675f20f	2016-02-20 14:22:52	2016-02-20 17:53:07.0	
1894	5d5d588e-81d8-41af-9072-907f774f265d	68d7e58c-9bda-4819-b624-cf48c1648c88	e14660b7-40df-424c-891e-9dc03c926fe1	2016-06-14 18:17:11	2016-06-14 18:37:09.0	
2173	82e1cd52-468f-4e9d-84b4-38e2c24ec430	be380032-86df-4f5c-a9fb-ffb598c66d73	4fcd4f1c-5886-4b48-b3b5-6fa5425584b2	2016-01-22 19:57:10	2016-01-23 02:52:14.0	

5 rows × 27 columns

In [70]:
```
#There is an issue of reviews for the listing that are negative. I will turn t
# Might change it to zero later.
# Normally, I would just look at actuall number of reviews in the listings (wh
contacts_all[contacts_all['id_listing_anon'] == '46ed561a-497c-4de8-b120-47f42
```

Out[70]:

	id_guest_anon	id_host_anon	id_listing_anon	ts_interaction_first	ts_reply_at_first	ts_acc
1282	8092c7ee-1bdc-4587-8aab-71268b6c6d1d	6c319423-9406-407d-a662-973862ee1e7c	46ed561a-497c-4de8-b120-47f42ffc9159	2016-03-20 16:56:24	2016-03-21 15:41:45.0	

In [71]:
```
# Turning it positive
contacts_all['total_reviews_pos'] = contacts_all['total_reviews'].abs()
```

In [72]:
```
#Mean and standard deviation barely moved
contacts_all.describe()
```

Out[72]:

	m_guests	m_interactions	m_first_message_length_in_characters	words_in_user_prof
count	27978.000000	27979.000000	27979.000000	27979.0000
mean	2.782436	8.247686	193.743879	10.2024
std	1.866441	10.025091	183.169421	30.8467
min	0.000000	1.000000	0.000000	0.0000
25%	2.000000	3.000000	66.000000	0.0000
50%	2.000000	5.000000	154.000000	0.0000
75%	4.000000	10.000000	269.000000	1.0000
max	16.000000	410.000000	2341.000000	658.0000

In [73]:
```
# Add a booking flag
contacts_all['book_flag'] = np.where(contacts_all['ts_booking_at'].isnull(), 0
```

In [74]:
```
contacts_all.describe()
```

Out[74]:

	m_guests	m_interactions	m_first_message_length_in_characters	words_in_user_prof
count	27978.000000	27979.000000	27979.000000	27979.0000
mean	2.782436	8.247686	193.743879	10.2024
std	1.866441	10.025091	183.169421	30.8467
min	0.000000	1.000000	0.000000	0.0000
25%	2.000000	3.000000	66.000000	0.0000
50%	2.000000	5.000000	154.000000	0.0000
75%	4.000000	10.000000	269.000000	1.0000
max	16.000000	410.000000	2341.000000	658.0000

In [75]:
```
import datetime as dt
```

In [ ]:
```

```

In [76]:
```
_days'] = (pd.to_datetime(contacts_all['ds_checkin_first']).dt.date - pd.to_dat
k_in_days'] = (pd.to_datetime(contacts_all['ds_checkin_first']).dt.date - pd.to
```

In [77]:
```
contacts_all['stay_duration_days'] = (pd.to_datetime(contacts_all['ds_checkout
contacts_all['interact_2_reply_days'] = (pd.to_datetime(contacts_all['ts_reply
```

In [229]:
```
datetime(contacts_all['ts_booking_at']).dt.date - pd.to_datetime(contacts_all['
```

In [230]:
```
# contacts_all[contacts_all['book_2_check_in_days'] < 0]
```

In [231]:
```
contacts_all.head()
```

Out[231]:

al_reviews	total_reviews_pos	book_flag	book_2_check_in_days	interact_2_check_in_days	stay_d
9.0	9.0	1	103.0	103	
0.0	0.0	0	NaN	177	
49.0	49.0	0	NaN	47	
17.0	17.0	0	NaN	22	
1.0	1.0	1	57.0	57	

In [232]:
```
contacts_all.describe()
```

Out[232]:

	m_guests	m_interactions	m_first_message_length_in_characters	words_in_user_prof
count	27978.000000	27979.000000	27979.000000	27979.0000
mean	2.782436	8.247686	193.743879	10.2024
std	1.866441	10.025091	183.169421	30.8467
min	0.000000	1.000000	0.000000	0.0000
25%	2.000000	3.000000	66.000000	0.0000
50%	2.000000	5.000000	154.000000	0.0000
75%	4.000000	10.000000	269.000000	1.0000
max	16.000000	410.000000	2341.000000	658.0000

In [233]:
```
contacts_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 27979 entries, 0 to 27978
Data columns (total 27 columns):
id_guest_anon                          27979 non-null object
id_host_anon                           27979 non-null object
id_listing_anon                        27979 non-null object
ts_interaction_first                   27979 non-null object
ts_reply_at_first                      25935 non-null object
ts_accepted_at_first                   16454 non-null object
ts_booking_at                          11608 non-null object
ds_checkin_first                       27979 non-null object
ds_checkout_first                      27979 non-null object
m_guests                               27978 non-null float64
m_interactions                         27979 non-null int64
m_first_message_length_in_characters   27979 non-null float64
contact_channel_first                  27979 non-null object
guest_user_stage_first                 27979 non-null object
id_user_anon                           27979 non-null object
country                                27978 non-null object
words_in_user_profile                  27979 non-null int64
room_type                              27979 non-null object
listing_neighborhood                   27979 non-null object
total_reviews                          27979 non-null float64
total_reviews_pos                      27979 non-null float64
book_flag                              27979 non-null int32
book_2_check_in_days                   11608 non-null float64
interact_2_check_in_days               27979 non-null int64
stay_duration_days                     27979 non-null int64
interact_2_reply_days                  25935 non-null float64
interact_2_book_in_days                11608 non-null float64
dtypes: float64(7), int32(1), int64(4), object(15)
memory usage: 7.1+ MB
```

In [241]:
```
# Negative book2_check_in (they booked after they arrived) (40 count)
# For the purpose of this excercise, I will remove these, however, I would loo
# contacts_all[(contacts_all['book_2_check_in_days'] < 0) ].count()
```

In [235]:
```
# Dropped the negative values but kept the nulls
contacts_all_v2 = contacts_all[((contacts_all['book_2_check_in_days'] >= 0) |
```

In [236]:
```
contacts_all_v2.describe()
```

Out[236]:

	m_guests	m_interactions	m_first_message_length_in_characters	words_in_user_prof
count	27934.000000	27934.000000	27934.000000	27934.0000
mean	2.784206	8.233264	193.844133	10.1987
std	1.866784	10.006696	183.208540	30.8463
min	1.000000	1.000000	0.000000	0.0000
25%	2.000000	3.000000	66.000000	0.0000
50%	2.000000	5.000000	154.000000	0.0000
75%	4.000000	10.000000	269.000000	1.0000
max	16.000000	410.000000	2341.000000	658.0000

In [237]:
```
#Quick Check
contacts_all_v2[contacts_all_v2['book_2_check_in_days'] < 0]
```

Out[237]:

id_guest_anon	id_host_anon	id_listing_anon	ts_interaction_first	ts_reply_at_first	ts_accepted

0 rows × 27 columns

In [ ]:
```

```

In [239]:
```
# Location Unknown Flag country
contacts_all_v2['location_unknown_flag'] = np.where(contacts_all_v2['ts_bookin
contacts_all_v2['br_country_flag'] = np.where(contacts_all_v2['country'] == 'B
contacts_all_v2['location_unknown_flag'] = np.where(contacts_all_v2['listing_n
contacts_all_v2['br_country_flag'] = np.where(contacts_all_v2['country'] == 'B
contacts_all_v2['room_apt_flag'] = np.where(contacts_all_v2['room_type'] == 'E
contacts_all_v2['room_private_flag'] = np.where(contacts_all_v2['room_type'] =
contacts_all_v2['room_shared_flag'] = np.where(contacts_all_v2['room_type'] ==
contacts_all_v2['new_user_flag'] = np.where(contacts_all_v2['guest_user_stage_
contacts_all_v2['contact_me_flag'] = np.where(contacts_all_v2['contact_channel
```

```
ocs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pydat
a.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
C:\Program Files\Anaconda3\lib\site-packages\ipykernel\__main__.py:5: S
ettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-d
ocs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pydat
a.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
C:\Program Files\Anaconda3\lib\site-packages\ipykernel\__main__.py:6: S
ettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-d
ocs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pydat
a.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
C:\Program Files\Anaconda3\lib\site-packages\ipykernel\__main__.py:7: S
ettingWithCopyWarning:
```

In [243]:
```
contacts_all_v2.describe()
```

Out[243]:

	m_guests	m_interactions	m_first_message_length_in_characters	words_in_user_prof
count	27934.000000	27934.000000	27934.000000	27934.0000
mean	2.784206	8.233264	193.844133	10.1987
std	1.866784	10.006696	183.208540	30.8463
min	1.000000	1.000000	0.000000	0.0000
25%	2.000000	3.000000	66.000000	0.0000
50%	2.000000	5.000000	154.000000	0.0000
75%	4.000000	10.000000	269.000000	1.0000
max	16.000000	410.000000	2341.000000	658.0000

Quick Save to CSV

In [244]:
```
# Quick Save:
contacts_all_v2.to_csv('contacts_all_v2.csv', index=False)
```

In [245]:
```
# And read the saved csv
contacts_all_v3 = pd.read_csv('contacts_all_v2.csv')
```

In [246]:
```
# contacts_all_v3.info()
```

Exploring Variables

In [271]:
```
contacts_all_v3[['contact_channel_first','book_flag']].groupby(['contact_chann
```

Out[271]:

contact_channel_first	book_flag
book_it	3973
contact_me	896
instant_book	6697

In [248]:
```
contacts_all_v3.groupby(['contact_channel_first']).count()
```

Out[248]:

contact_channel_first	id_guest_anon	id_host_anon	id_listing_anon	ts_interaction_first	ts_reply_
book_it	8364	8364	8364	8364	
contact_me	12873	12873	12873	12873	
instant_book	6697	6697	6697	6697	

3 rows × 33 columns

In [249]:
```
contacts_all_v3.groupby(['contact_channel_first']).median()
```

Out[249]:

contact_channel_first	m_guests	m_interactions	m_first_message_length_in_characters	words_i
book_it	2.0	5	125.0	
contact_me	2.0	4	183.0	
instant_book	2.0	9	122.0	

In [250]:
```
contacts_all_v3.groupby(['contact_channel_first']).mean()
```

Out[250]:

	location_unknown_flag	br_country_flag	room_apt_flag	room_private_flag	room_shared_flag	ne
)	0.412363	0.414395	0.735175	0.242348	0.022477	
3	0.464694	0.615863	0.777441	0.198244	0.024314	
3	0.361356	0.341795	0.728087	0.258773	0.013140	

In [107]:
```
contacts_all_v2.groupby(['listing_neighborhood','contact_channel_first']).coun
```

201	201	201	201	201	201
1	1	1	1	1	0
4	4	4	4	4	0
1	1	1	1	1	0
1	1	1	1	1	1
1	1	1	1	1	0
2	2	2	2	2	0
257	257	257	257	257	126
339	339	339	339	339	20
159	159	159	159	159	159
6	6	6	6	6	0

In [ ]:
```
```

In [218]:
```
# Flags
# contacts_all_v3['location_unknown_flag'] = np.where(contacts_all_v3['listing
# contacts_all_v3['br_country_flag'] = np.where(contacts_all_v3['country'] ==
```

In [251]:
```
contacts_all_v3.describe()
```

Out[251]:

	m_guests	m_interactions	m_first_message_length_in_characters	words_in_user_prof
count	27934.000000	27934.000000	27934.000000	27934.0000
mean	2.784206	8.233264	193.844133	10.1987
std	1.866784	10.006696	183.208540	30.8463
min	1.000000	1.000000	0.000000	0.0000
25%	2.000000	3.000000	66.000000	0.0000
50%	2.000000	5.000000	154.000000	0.0000
75%	4.000000	10.000000	269.000000	1.0000
max	16.000000	410.000000	2341.000000	658.0000

In [253]:
```
contacts_all_v3.groupby(['new_user_flag','location_unknown_flag','contact_chan
```

Out[253]:

			id_guest_anon	id_host_anon	id_lis
new_user_flag	**location_unknown_flag**	**contact_channel_first**			
0	**0**	**book_it**	2454	2454	
		contact_me	2263	2263	
		instant_book	2553	2553	
	1	**book_it**	1609	1609	
		contact_me	1780	1780	
		instant_book	1341	1341	
1	**0**	**book_it**	2461	2461	
		contact_me	4628	4628	
		instant_book	1724	1724	
	1	**book_it**	1840	1840	
		contact_me	4202	4202	
		instant_book	1079	1079	

12 rows × 31 columns

How do you move new users from Contact me to book_it?

In [254]:
```
contacts_all_v3.groupby(['new_user_flag','br_country_flag','location_unknown_f
```

Out[254]:

				id_guest_anon	id_
new_user_flag	br_country_flag	location_unknown_flag	contact_channel_first		
0	0	0	book_it	1738	
			contact_me	1176	
			instant_book	1967	
		1	book_it	1093	
			contact_me	850	
			instant_book	937	
	1	0	book_it	716	
			contact_me	1087	
			instant_book	586	
		1	book_it	516	
			contact_me	930	
			instant_book	404	
1	0	0	book_it	1221	
			contact_me	1595	
			instant_book	957	
		1	book_it	846	
			contact_me	1324	
			instant_book	547	
	1	0	book_it	1240	
			contact_me	3033	
			instant_book	767	
		1	book_it	994	
			contact_me	2878	
			instant_book	532	

24 rows × 30 columns

In [204]:
```
contacts_all_v3.groupby(['new_user_flag','br_country_flag']).count()
```

Out[204]:

new_user_flag	br_country_flag	id_guest_anon	id_host_anon	id_listing_anon	ts_interaction_first
0	0	7761	7761	7761	7761
	1	4239	4239	4239	4239
1	0	6490	6490	6490	6490
	1	9444	9444	9444	9444

4 rows × 30 columns

In [ ]:
```
# Non-brasilians have more interactions (ask more questions)
# Their messages are longer
# Their profiles are longer
# Seems like they go for places with more reviews
#
```

In [205]:
```
contacts_all_v3.groupby(['new_user_flag','br_country_flag']).mean()
```

Out[205]:

new_user_flag	br_country_flag	m_guests	m_interactions	m_first_message_length_in_characters
0	0	2.379461	9.896147	221.20448
	1	2.815994	7.382166	136.19745
1	0	2.796610	8.533744	238.80662
	1	3.094028	7.042249	166.33608

In [ ]:
```
#Seems that brasilians have lower booking ratios
```

In [228]:
```
contacts_all_v3.groupby(['new_user_flag','br_country_flag','contact_me_flag'])
```

Out[228]:

new_user_flag	br_country_flag	contact_me_flag	m_guests	m_interactions	m_first_message_le
0	0	0	2.284568	11.002092	
		1	2.648075	6.765548	
	1	0	2.622412	8.771827	
		1	3.029251	5.851264	
1	0	0	2.613274	10.596752	
		1	3.020898	6.009935	
	1	0	2.890744	8.863289	
		1	3.215530	5.953815	

In [163]:
```
contacts_all_v3.groupby(['br_country_flag','location_unknown_flag','contact_ch
```

Out[163]:

br_country_flag	location_unknown_flag	contact_channel_first	m_guests	m_interactions	m_first
0	0	book_it	2.478202	9.415343	
		contact_me	2.858174	6.249729	
		instant_book	2.362859	12.293434	
	1	book_it	2.428571	9.227953	
		contact_me	2.880865	6.408464	
		instant_book	2.347035	12.964286	
1	0	book_it	2.864008	8.221881	
		contact_me	3.164078	5.878883	
		instant_book	2.659276	10.133777	
	1	book_it	2.896026	7.943046	
		contact_me	3.172532	5.980567	
		instant_book	2.635684	9.634615	

In [257]:
```
contacts_all_v3.groupby(['listing_neighborhood']).mean()
```

Out[257]:

listing_neighborhood	m_guests	m_interactions	m_first_message_length_in_characters	words_in

listing_neighborhood			
-unknown-	2.816302	8.003544	192.001603
Abolição	3.000000	5.000000	123.000000
Andaraí	2.000000	7.333333	176.500000
Barra da Tijuca	3.045908	7.978044	178.698603
Barros Filho	2.000000	7.800000	147.400000
Benfica	2.000000	4.500000	66.000000
Bonsucesso	1.666667	2.333333	188.000000
Botafogo	2.298013	7.864901	212.150993
Brás de Pina	3.833333	5.000000	126.833333
Catete	2.259259	8.166667	212.814815
Catumbi	2.250000	12.250000	42.000000
Centro	1.991736	8.198347	215.314050
Cidade Nova	1.250000	6.500000	348.500000
Coelho Neto	2.000000	3.000000	0.000000
Copacabana	2.919051	8.426779	197.037184
Cosme Velho	2.210526	9.552632	165.921053
Del Castilho	2.333333	9.333333	148.666667
Encantado	3.000000	4.500000	140.000000
Engenho Novo	1.250000	6.750000	285.750000
Engenho da Rainha	4.000000	1.000000	120.000000
Engenho de Dentro	2.416667	6.833333	133.416667
Estacio	2.735294	7.029412	158.470588
Flamengo	2.300654	8.736383	181.348584
Gamboa	1.000000	5.800000	137.800000
Glória	2.539062	9.125000	215.031250
Grajaú	2.153846	6.692308	270.769231
Guadalupe	3.000000	2.000000	0.000000
Gávea	3.183908	11.643678	220.241379
Humaitá	2.500000	8.100000	187.266667
Ipanema	2.853000	8.212333	189.160000
...	...	...	...
Madureira	1.000000	16.000000	324.000000
Maracanã	2.250000	6.218750	156.656250
Marechal Hermes	1.000000	5.000000	490.000000

Méier	1.800000	5.200000	216.500000
Olaria	1.000000	7.000000	26.000000
Parada de Lucas	1.000000	19.000000	88.400000
Penha	1.500000	6.875000	156.625000
Penha Circular	6.000000	3.375000	154.875000
Praça da Bandeira	2.925926	9.333333	192.851852
Quintino Bocaiúva	5.000000	8.000000	274.000000
Ramos	1.800000	7.200000	92.000000
Recreio dos Bandeirantes	2.995763	8.800847	192.398305
Riachuelo	2.000000	7.333333	153.000000
Rio Comprido	2.333333	6.666667	243.962963
Rocha	2.750000	13.000000	163.250000
Rocinha	1.000000	1.000000	188.000000
Sampaio	1.000000	4.000000	8.000000
Santa Teresa	2.035775	9.175468	228.744463
Santo Cristo	2.727273	7.272727	197.909091
Saúde	1.500000	9.000000	122.500000
São Conrado	3.500000	8.800000	217.625000
São Cristóvão	2.941176	5.529412	238.000000
Tijuca	1.963303	8.174312	233.266055
Todos os Santos	1.600000	12.000000	147.200000
Urca	2.384615	12.051282	216.192308
Vaz Lobo	1.000000	1.000000	120.000000
Vidigal	2.201923	9.384615	272.038462
Vigário Geral	2.000000	2.000000	329.000000
Vila Isabel	2.274510	8.529412	190.117647
Vila da Penha	1.000000	1.000000	820.000000

68 rows × 19 columns

Overall, brasilians contact more

In [165]:
```
contacts_all_v3.groupby(['br_country_flag','room_type','contact_channel_first'
```

Out[165]:

br_country_flag	room_type	contact_channel_first	id_guest_anon	id_host_anon	id_listing_anon	t
0	Entire home/apt	book_it	3479	3479	3479	
		contact_me	3791	3791	3791	
		instant_book	3151	3151	3151	
	Private room	book_it	1303	1303	1303	
		contact_me	1027	1027	1027	
		instant_book	1202	1202	1202	
	Shared room	book_it	116	116	116	
		contact_me	127	127	127	

In [130]:
```
contacts_all_v2.groupby(['room_type']).mean()
```

Out[130]:

room_type	m_guests	m_interactions	m_first_message_length_in_characters	words_in_user_prc
Entire home/apt	3.128120	8.291229	188.203043	9.138
Private room	1.696024	8.104213	212.114349	13.736
Shared room	2.147458	7.535593	199.030508	10.067

Quickly explore if Brasilians book more than 1 place and then cancel

The answer is that the value is very similar and I would not attribute it to that

In [206]:
```
from pandasql import sqldf
```

In [207]:
```
pysqldf = lambda q: sqldf(q, globals())
```

In [215]:
```
q = """SELECT  id_guest_anon, ds_checkin_first, max(br_country_flag) br_countr
FROM contacts_all_v3 group by id_guest_anon, ds_checkin_first """
abc = pysqldf(q)
```

In [ ]:
```
q2 = """SELECT br_country_flag, m_count, count(*) \
FROM abc group by br_country_flag,  m_count order by br_country_flag,  m_count
pysqldf(q2)
```

In [217]:
```
q2 = """SELECT br_country_flag, case when m_count = 1 then 0 else 1 end more_1
FROM abc group by br_country_flag,  case when m_count = 1 then 0 else 1 end or
pysqldf(q2)
```

Out[217]:

	br_country_flag	more_1	count(*)
0	0	1	1373
1	0	0	11038
2	1	1	1568
3	1	0	9890

In [293]:
```
q3 = """SELECT book_flag, br_country_flag, count(*) \
FROM contacts_all_v3 where words_in_user_profile = 0 and new_user_flag = 0 and
pysqldf(q3)
```

Out[293]:

	book_flag	br_country_flag	count(*)
0	0	0	673
1	0	1	897
2	1	0	79
3	1	1	70

In [292]:
```
q4 = """SELECT book_flag, br_country_flag, count(*) \
FROM contacts_all_v3 where new_user_flag = 0 and contact_me_flag = 1 group by
pysqldf(q4)
```

Out[292]:

	book_flag	br_country_flag	count(*)
0	0	0	1808
1	0	1	1871
2	1	0	218
3	1	1	146

Model

In [118]:
```
from sklearn import preprocessing
from sklearn.linear_model import LogisticRegression
from sklearn.cross_validation import train_test_split
from sklearn import metrics
from sklearn.metrics import classification_report
```

```
C:\Program Files\Anaconda3\lib\site-packages\sklearn\cross_validation.py:
41: DeprecationWarning: This module was deprecated in version 0.18 in fav
or of the model_selection module into which all the refactored classes an
d functions are moved. Also note that the interface of the new CV iterato
rs are different from that of this module. This module will be removed in
 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
```

In [286]:
```
contacts_all_v3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27934 entries, 0 to 27933
Data columns (total 34 columns):
id_guest_anon                              27934 non-null object
id_host_anon                               27934 non-null object
id_listing_anon                            27934 non-null object
ts_interaction_first                       27934 non-null object
ts_reply_at_first                          25890 non-null object
ts_accepted_at_first                       16412 non-null object
ts_booking_at                              11566 non-null object
ds_checkin_first                           27934 non-null object
ds_checkout_first                          27934 non-null object
m_guests                                   27934 non-null float64
m_interactions                             27934 non-null int64
m_first_message_length_in_characters       27934 non-null float64
contact_channel_first                      27934 non-null object
guest_user_stage_first                     27934 non-null object
id_user_anon                               27934 non-null object
country                                    27933 non-null object
words_in_user_profile                      27934 non-null int64
room_type                                  27934 non-null object
listing_neighborhood                       27934 non-null object
total_reviews                              27934 non-null float64
total_reviews_pos                          27934 non-null float64
book_flag                                  27934 non-null int64
book_2_check_in_days                       11566 non-null float64
interact_2_check_in_days                   27934 non-null int64
stay_duration_days                         27934 non-null int64
interact_2_reply_days                      25890 non-null float64
interact_2_book_in_days                    11566 non-null float64
location_unknown_flag                      27934 non-null int64
br_country_flag                            27934 non-null int64
room_apt_flag                              27934 non-null int64
room_private_flag                          27934 non-null int64
room_shared_flag                           27934 non-null int64
new_user_flag                              27934 non-null int64
contact_me_flag                            27934 non-null int64
dtypes: float64(7), int64(12), object(15)
memory usage: 7.2+ MB
```

What Impacts Booking?

All Users

In [219]:
```
X =  contacts_all_v3[['contact_me_flag','new_user_flag','br_country_flag','loc
y = contacts_all_v3['book_flag']
```

In [220]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = .3, rand
```

In [221]:
```
total = X.isnull().sum().sort_values(ascending = False)
percent = (X.isnull().sum()/X.isnull().count()*100).sort_values(ascending = Fa
missing_contacts  = pd.concat([total, percent], axis=1, keys=['Total', 'Percen
missing_contacts.head(20)
```

Out[221]:

	Total	Percent
stay_duration_days	0	0.0
interact_2_check_in_days	0	0.0
total_reviews_pos	0	0.0
room_private_flag	0	0.0
room_apt_flag	0	0.0
words_in_user_profile	0	0.0
m_guests	0	0.0
m_first_message_length_in_characters	0	0.0
m_interactions	0	0.0
location_unknown_flag	0	0.0
br_country_flag	0	0.0
new_user_flag	0	0.0
contact_me_flag	0	0.0

In [222]:
```
LogReg = LogisticRegression()
LogReg.fit(X_train, y_train)
```

Out[222]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=Tr
ue,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001
,
          verbose=0, warm_start=False)
```

In [224]:
```
y_pred = LogReg.predict(X_test)
```

In [225]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
confusion_matrix
```

Out[225]:
```
array([[4231,  598],
       [ 506, 3046]], dtype=int64)
```

In [226]:
```
print(classification_report(y_test, y_pred))
```

```
             precision    recall  f1-score   support

          0       0.89      0.88      0.88      4829
          1       0.84      0.86      0.85      3552

avg / total       0.87      0.87      0.87      8381
```

In [227]:
```
# examine the coefficients
pd.DataFrame(list(zip(X.columns, np.transpose(LogReg.coef_))))
```

Out[227]:

	0	1
0	contact_me_flag	[-3.6916352088130915]
1	new_user_flag	[-0.17481105582023848]
2	br_country_flag	[-0.4091219692434835]
3	location_unknown_flag	[-0.027358793098680823]
4	m_interactions	[0.2123524918073225]
5	m_first_message_length_in_characters	[-0.0005528026690888459]
6	m_guests	[-0.048911079777550534]
7	words_in_user_profile	[0.0002113149186211171]
8	room_apt_flag	[0.042928435199020246]
9	room_private_flag	[0.12936798408461705]
10	total_reviews_pos	[0.018582926382815314]
11	interact_2_check_in_days	[-0.004755524170973826]
12	stay_duration_days	[-0.05205050359092526]

What defines a Brazilian?

In [258]:
```
X2 =  contacts_all_v3[['book_flag','contact_me_flag','new_user_flag','location
y2 = contacts_all_v3['br_country_flag']
```

In [259]:
```
X_train2, X_test2, y_train2, y_test2 = train_test_split(X2, y2, test_size = .3
```

In [260]:
```
LogReg2 = LogisticRegression()
LogReg2.fit(X_train2, y_train2)
```

Out[260]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=Tr
ue,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001
,
          verbose=0, warm_start=False)
```

In [263]:
```
y_pred2 = LogReg2.predict(X_test2)
```

In [265]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix2 = confusion_matrix(y_test2, y_pred2)
confusion_matrix2
```

Out[265]:
```
array([[2942, 1353],
       [1282, 2804]], dtype=int64)
```

In [267]:
```
print(classification_report(y_test2, y_pred2))
```

```
             precision    recall  f1-score   support

          0       0.70      0.68      0.69      4295
          1       0.67      0.69      0.68      4086

avg / total       0.69      0.69      0.69      8381
```

In [268]:
```
# examine the coefficients
pd.DataFrame(list(zip(X2.columns, np.transpose(LogReg2.coef_))))
```

Out[268]:

	0	1
0	book_flag	[-0.34201324930188415]
1	contact_me_flag	[0.8519732109502396]
2	new_user_flag	[0.8064358076108182]
3	location_unknown_flag	[0.10082447292492748]
4	m_interactions	[-0.007633085505490978]
5	m_first_message_length_in_characters	[-0.0032097730588358606]
6	m_guests	[0.057998247521943554]
7	words_in_user_profile	[-0.004507536320170442]
8	room_apt_flag	[0.19906801050853667]
9	room_private_flag	[0.1344828056317339]
10	total_reviews_pos	[-0.004954566784209933]
11	interact_2_check_in_days	[-0.002051140877476678]
12	stay_duration_days	[-0.013971190160498262]

Keep just Brazilians

Check the booking impact

In [272]:
```
contacts_all_v3_br = contacts_all_v3[contacts_all_v3['br_country_flag'] == 1]
```

In [275]:
```
X3 =  contacts_all_v3_br[['contact_me_flag','new_user_flag','location_unknown_
y3 = contacts_all_v3_br['book_flag']
```

In [276]:
```
X_train3, X_test3, y_train3, y_test3 = train_test_split(X3, y3, test_size = .3
```

In [277]:
```
LogReg3 = LogisticRegression()
LogReg3.fit(X_train3, y_train3)
```

Out[277]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=Tr
ue,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001
,
          verbose=0, warm_start=False)
```

In [278]:
```
y_pred3 = LogReg3.predict(X_test3)
```

In [280]:
```
confusion_matrix3 = confusion_matrix(y_test3, y_pred3)
confusion_matrix3
```

Out[280]:
```
array([[2567,  287],
       [ 237, 1014]], dtype=int64)
```

In [281]:
```
print(classification_report(y_test3, y_pred3))
```

```
             precision    recall  f1-score   support

          0       0.92      0.90      0.91      2854
          1       0.78      0.81      0.79      1251

avg / total       0.87      0.87      0.87      4105
```

In [282]:
```
# examine the coefficients
pd.DataFrame(list(zip(X3.columns, np.transpose(LogReg3.coef_))))
```

Out[282]:

	0	1
0	contact_me_flag	[-3.727980057086825]
1	new_user_flag	[-0.2802596825210466]
2	location_unknown_flag	[0.007394300562890962]
3	m_interactions	[0.16178508753142082]
4	m_first_message_length_in_characters	[-0.00027934023201824586]
5	m_guests	[-0.0677917224496325]
6	words_in_user_profile	[-0.0003239430472334989]
7	room_apt_flag	[0.04317297035229868]
8	room_private_flag	[0.15169835593109418]
9	total_reviews_pos	[0.016152795701454573]
10	interact_2_check_in_days	[-0.004778881598572779]
11	stay_duration_days	[-0.04807451054195284]

In [ ]:

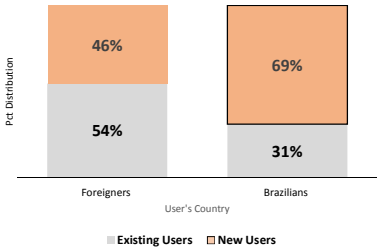
In [ ]:

In [ ]:

Column Labels						
	0			1		
Row Labels	Count of id_guest_anon	Cour	Average of book_flag	Cou	Count of id_host_anon	Average of book_flag
0	7761	###	59.3%	##	45.5%	42.3%
1	4239	###	39.8%	##	69.0%	26.8%

Distribution of Guests

	Existing Users	New Users
Foreigners	54%	###
Brazilians	31%	###



		Conversion Rate *
	Foreign	Brazilians
New Users	##	27%
Existing Users	##	40%

* Booking Count / First Contact Count

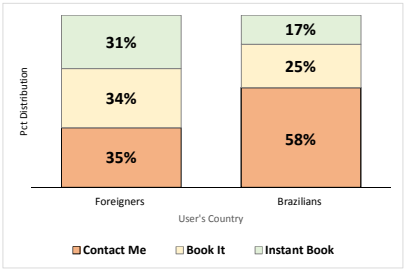
Column Labels							
	contact_me		book_it		instant_book		
Row Labels	Count of id_host_anon	Aver	Count of id_host_anon	Ave	Count of id_host_anon		Average of book_flag
0	34.70%	0.1	34.37%	1	30.93%		1
1	57.94%	0.1	25.33%	0	16.73%		1

User Distribution

Contact Me		Book	Instant Book
Foreigners	35%	###	31%
Brazilians	58%	###	17%

Conversion Rate *

	Contact Me	ook I	Instant Book
Foreigners	8.7%	###	100.0%
Brazilians	5.9%	###	100.0%



	Conversion Rate *	
	Foreigners	Brazilians
Book It	9%	6%
Instant Book	51%	42%
	100%	100%

* Booking Count / First Contact Count

	Foreigners	Brazilians
Contact	8.7%	##
Book I	51.2%	##
Instant B	100.0%	##

contact_channel_first (All)

Row Labels	Column Labels: 0 — Count of id_guest_anon	Sum Count of id_guest_anon	1 — Sum of book_flag	
1	46%	###	69%	##
0	54%	###	31%	##

"Contact Me"

	Foreigners		Brazilians	
	Contact Share	Booking Share	Contact Share	Booking Share
New User	59%	50%	**75%**	**69%**
Existing User	41%	50%	**25%**	**31%**

contact_channel_first contact_me

Average of m_first_me	Column Labels: 0	1
Row Labels		
0		
1	268	206
0	267	192
1		
1	307	267
0	294	229

	Word Count Of The First Message	
	Foreigners	Brazilians
Failed Booking		
New User	268	206
Existing User	267	192
Successful Booking		
New User	307	267
Existing User	294	229
Successful - Failed Booking		
New User	**40**	**61**
Existing User	**27**	**37**
New User (% Dif)	**15%**	**30%**
Existing User (% Dif)	**10%**	**19%**

	New User				
		3110	2111		
		7761	4239		
		40.1%	49.8%		
BR	0	897	1871	48%	
	1	70	146	48%	
F	0	673	1808	37%	
	1	79	218	36%	

contact_channel_first contact_me
new_user_flag 1

Row Labels	Average of m_first_message_length_in_characters: 0	1	Average of words_in_user_profile: 0	1	Average of interact_2_check_in_days: 0	1	Average of total_reviews_pos: 0	1
0	268	307	7	10	101	82	9	18
1	206	267	4	3	78	63	9	14

Row Labels	erage Leangth of First Message (characte: eign	Brazilians	Average User Profile Length (words): eign	Brazilians
New User	206	267	4	3
Existing User	268	307	7	10

"Contact Me" and New Users Only

	Avg. Len. of First Message (characters)		Avg. Len. Of User Profile (words)		Avg. Num. Of Listing Reviews	
	No Booking	Booked	No Booking	Booked	No Booking	Booked
Brazilians	206.0	267.3	3.8	2.7	8.7	17.6
Foreigners	267.9	307.4	7.0	9.7	9.0	14.3
Difference	**-61.9**	**-40.2**	**-3.2**	**-7.0**	**-0.3**	**3.3**