Data Challenge Report

Overview

- Models for identifying factors that affect booking rate
  - Number of interactions, number of guests the inquiry is for, total number of reviews of a listing, time difference between checkin date and inquiry date, number of characters in the first message sent, number of days guests plan to stay, time difference between the first reply by host and first message sent by guest, room capacity
  - Product features that increase the blue variables and/or decrease red variables will likely increase booking rate
- Test of imposing word count requirement on first inquiry messages
  - We do not recommend to launch this feature to everyone.
  - However, further investigation might be needed for a deep dive of the effects on different population as well as its side effects.
- Experiment design for testing the change of response time limit

Identifying influential factors for booking rate and evaluating their effect sizes

Problem setup

- Given the study period,

$$\text{booking rate} = \frac{\text{number of total booking made}}{\text{number of total inquiries made}}$$

  - By its definition, booking rate = 100% for bookings made through '*instant_booked*' channel
  - Investigate what variables affect booking rate for bookings made via '*booked_it*' channel
- Classify whether a case to a booking or not based on its final booking status
  - If a case's associated *booking_at* timestamp is not NA, then it is a booking
- Logistic regression model
  - Assume the given variables have a linear relationship with the target variable 'booking or not'

Modeling: regressors

- Input variables considered initially
  - m_guests_first
  - m_guests_first m_interactions
  - m_first_message_length_in_characters
  - dim_room_type
  - dim_total_reviews
  - dim_person_capacity
  - dim_host_language
  - m_stay_days: *days between check out date ds_checkout_first and check in date ds_checkin_first*
  - m_interaction_to_checkin_days: *days between check in date ds_checkin_first and first inquiry made date ts_interaction_first*
  - m_interaction_to_reply_hours: *hours between first inquiry made time to first reply received time*

* Derived variables

Modeling: model fitting

```
Coefficients:
                                        Estimate Std. Error z value Pr(>|z|)
(Intercept)                           -1.1460637  0.3186816  -3.596 0.000323 ***
m_guests_first                         0.0262324  0.0167794   1.563 0.117966
m_interactions                         0.3636278  0.0084549  43.008  < 2e-16 ***
m_first_message_length_in_characters -0.0005744  0.0001129  -5.086 3.65e-07 ***
dim_room_typePrivate room            -0.5611509  0.0458756 -12.232  < 2e-16 ***
dim_room_typeShared room              0.0134750  0.1763985   0.076 0.939109
dim_total_reviews                     0.0156710  0.0006142  25.513  < 2e-16 ***
dim_person_capacity                  -0.1090111  0.0095264 -11.443  < 2e-16 ***
dim_host_languageen                   0.0805961  0.3203500   0.252 0.801360
dim_host_languagees                   0.0637729  0.3119962   0.204 0.838039
dim_host_languagefr                  -0.3521083  0.3736043  -0.942 0.345955
dim_host_languageit                   0.0611943  0.5399081   0.113 0.909759
dim_host_languagenl                  -0.4627842  1.2599450  -0.367 0.713391
dim_host_languagept                   0.4871799  0.8412704   0.579 0.562522
m_stay_days                          -0.0044715  0.0024948  -1.792 0.073080 .
m_interaction_to_checkin_days         0.0016395  0.0005684   2.885 0.003920 **
m_interaction_to_reply_hours         -0.0085199  0.0011122  -7.660 1.85e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- The results of first model run imply that a few variables’ coefficient estimates are not statistically significant
- Taking into account the correlations between the predictors, they should not be simply removed
- Apply a simple method like stepwise regression based on AIC criterion to do model selection

Modeling: model fitting

- Predictors in final model (removed *host language* from the initial model)
  - m_guests_first
  - m_guests_first m_interactions
  - m_first_message_length_in_characters
  - dim_room_type
  - dim_total_reviews
  - dim_person_capacity
  - ~~dim_host_language~~
  - m_stay_days: *days between check out date ds_checkout_first and check in date ds_checkin_first*
  - m_interaction_to_checkin_days: *days between check in date ds_checkin_first and first inquiry made date ts_interaction_first*
  - m_interaction_to_reply_hours: *hours between first inquiry made time to first reply received time*
- Limitation: not taking into account interaction terms between room type and other variables, therefore assume the effects of these variables are the same for different room types.

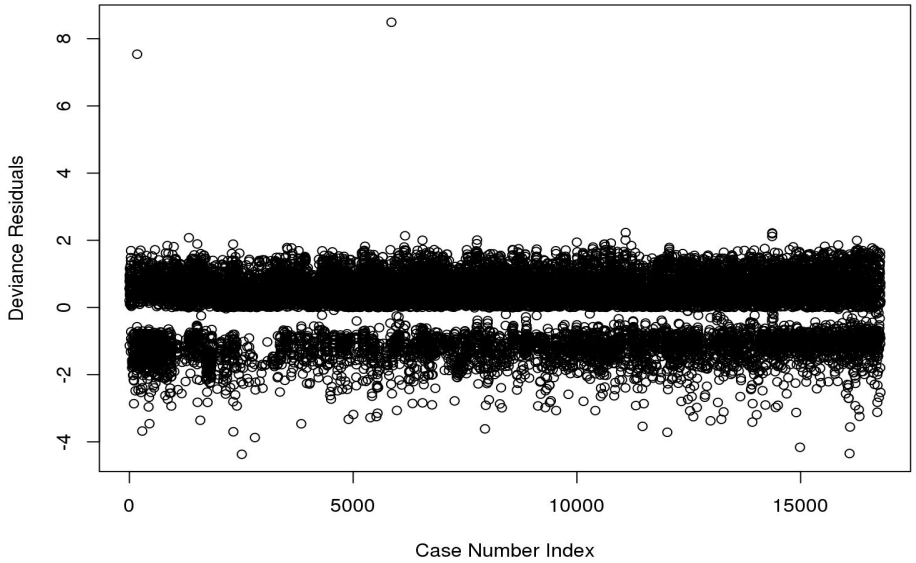
Modeling: model fitting

```
Call:
glm(formula = booking ~ m_guests_first + m_interactions + m_first_message_length_in_characters +
    dim_room_type + dim_total_reviews + dim_person_capacity +
    m_stay_days + m_interaction_to_checkin_days + m_interaction_to_reply_hours,
    family = binomial(link = logit), data = contactDT[dim_contact_channel_first ==
        "book_it"])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.3704  -0.7541   0.3216   0.7098   8.4904

Coefficients:
                                   Estimate Std. Error z value Pr(>|z|)
(Intercept)                      -1.0839989  0.0743507 -14.580  < 2e-16 ***
m_guests_first                    0.0262058  0.0167756   1.562  0.11826
m_interactions                    0.3634100  0.0084501  43.006  < 2e-16 ***
m_first_message_length_in_characters -0.0005750  0.0001129  -5.095 3.48e-07 ***
dim_room_typePrivate room        -0.5585493  0.0458001 -12.195  < 2e-16 ***
dim_room_typeShared room          0.0239599  0.1748407   0.137  0.89100
dim_total_reviews                 0.0156546  0.0006100  25.665  < 2e-16 ***
dim_person_capacity              -0.1092197  0.0095190 -11.474  < 2e-16 ***
m_stay_days                      -0.0044452  0.0024946  -1.782  0.07476 .
m_interaction_to_checkin_days     0.0016561  0.0005684   2.914  0.00357 **
m_interaction_to_reply_hours     -0.0085507  0.0011096  -7.706 1.30e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 19681  on 16784  degrees of freedom
Residual deviance: 14381  on 16774  degrees of freedom
  (3073 observations deleted due to missingness)
AIC: 14403
```

← Summary of final model output

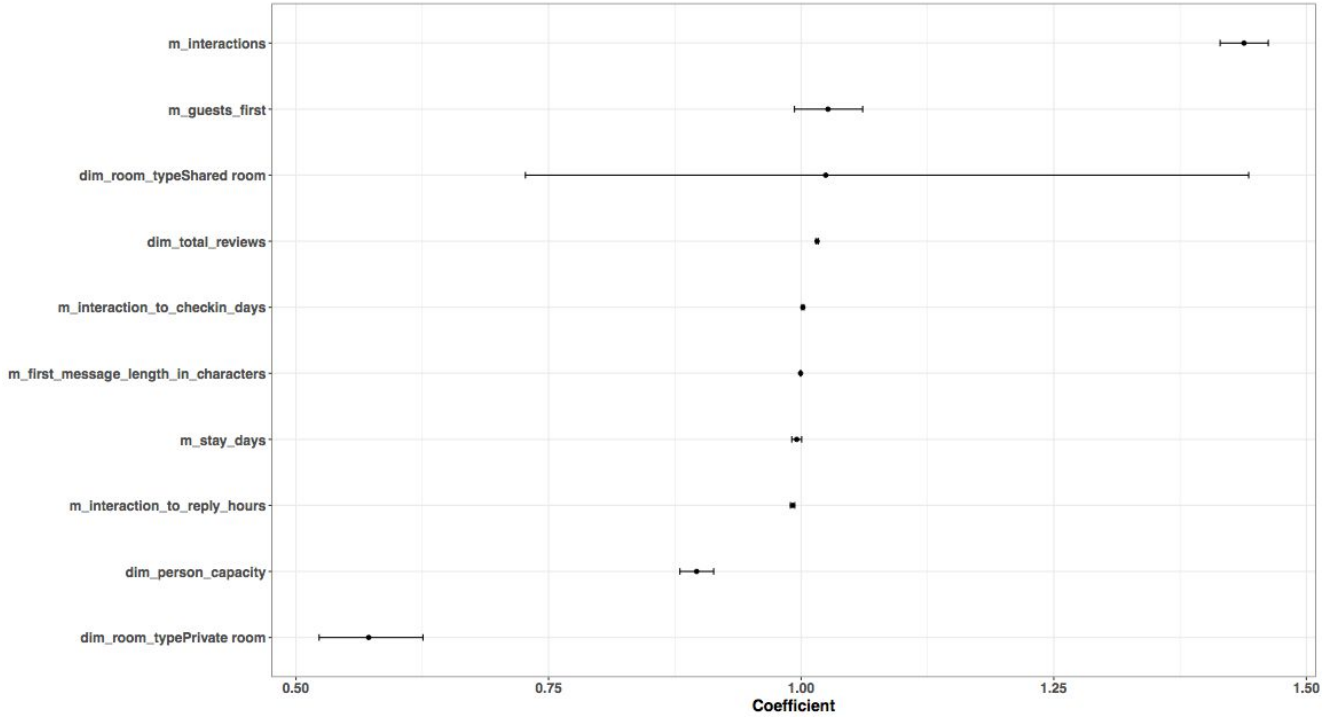
Modeling: model diagnostic

- Overall fitting is good. However, as the plot below shows, several deviance residuals fall out of the [-3, +3] bound, which needs further investigation.



Modeling: interpretation

- Exponentiate estimated coefficients (and corresponding 95% Wald confidence intervals) to get odds scale



Modeling: interpretation

- When holding other variables constant:
  - Increasing one time of interaction between guests and hosts can result in 1.44 times increase in terms of odds of booking over not booking.
  - Likewise, an inquiry is for more guests, a listing has more reviews, or an inquiry is for a booking for a closer coming days, are more likely ended up a booking.
  - For room types, *entire home/apt* and *shared room* do not have significant differences regarding booking rate. While comparing with them, *private room* has smaller chance of having a booking.
  - Longer lengths of stay, bigger person capacity listings, longer waiting time for the first replies, and more words in the first messages, all lead to booking rate decrease.
- Although these effects are evident in statistical sense, their practical change effects are judgement call.

Testing effects of imposing minimum 140 word requirement

Invariance check

- After cleaning the assignment list and contact data, for the cases in the contact data, 10145 and 10195 records are assigned to control and treatment group respectively.
- After the assignment, check if the other variables/attributes that are not affected by this word count requirement are invariant in two groups
  - Check the counts/proportions of different levels for a categorical variable are distributed roughly the same over two groups.

Contact channel

dim_contact_channel_first	group	N
book_it	control	8728
book_it	treatment	8763
instant_booked	control	1417
instant_booked	treatment	1432

Room type

dim_room_type	group	N
Private room	control	3052
Private room	treatment	3209
Entire home/apt	treatment	6853
Entire home/apt	control	6959
Shared room	control	134
Shared room	treatment	133

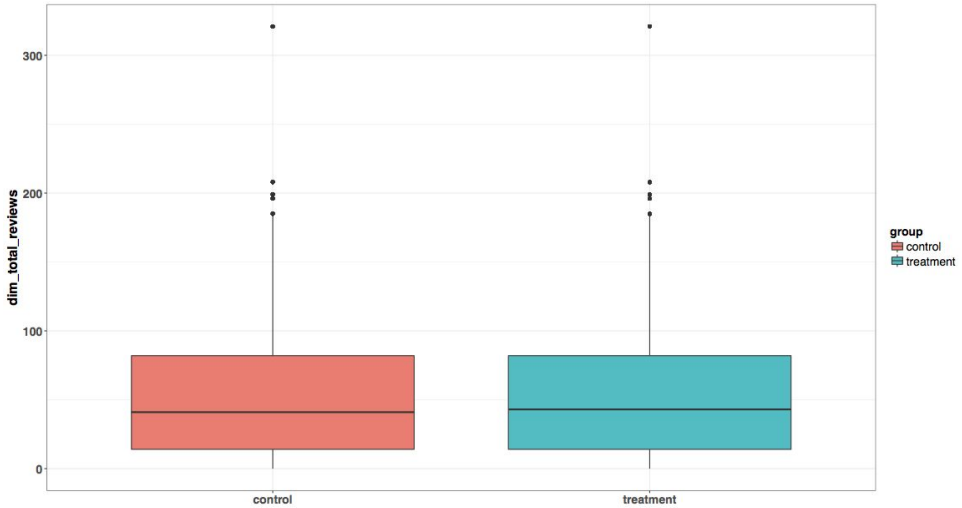
Host language

dim_host_language	group	N
es	control	9200
es	treatment	9217
fr	control	113
fr	treatment	126
de	control	48
de	treatment	45
en	control	758
en	treatment	777
ru	treatment	4
nl	treatment	2
it	control	17
pt	control	6
it	treatment	16
pt	treatment	8
nl	control	2
da	control	1

Invariance check

- Check the distributions of a numerical variable are approximately the same over two groups.

Example: number of total reviews



- Conclusion: for all the variables that are non-related to the test, there are no significant discrepancies between the two testing groups.

Testing

- Estimated booking rates (proportions of booked over all events)

```
      group booking_rate
    control    0.7298176
  treatment    0.7225110
```

- Using normal approximation and analytic estimates of the standard errors of the proportions, the z-test results are shown below

```
X-squared = 1.3652, df = 1, p-value = 0.2426
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.004949352  0.019562570
sample estimates:
   prop 1    prop 2
0.7298176 0.7225110
```

Results

- Since the difference is not statistical significant, we do not recommend to launch this feature to everyone.
- However, there might be evident differences among particular population segments. For instance, the population could be sliced based on gender, country, season, etc.
- Also, this change would lead to people being more involved in interactions, as a result of which booking rate increases.
- Therefore further investigation could be conducted.

Designing experiments for changing response time limit

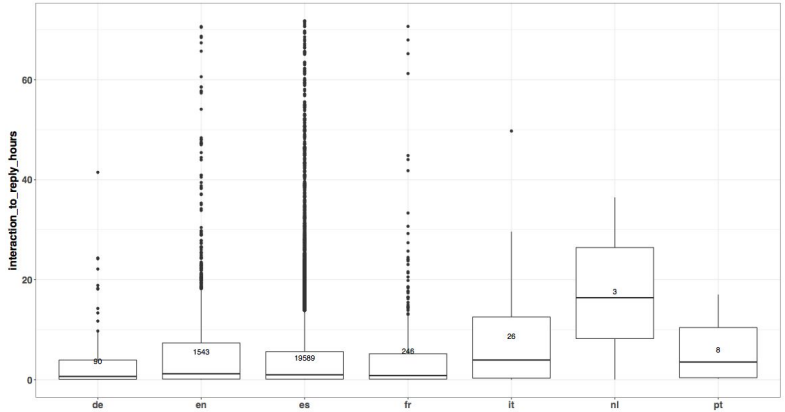
Pre-experiment thinking

- **How might the new time limit change the guest and host behavior?**
  - Hosts would be more actively involved in responding inquiries.
  - Hosts who do not have enough free time for replying inquiries might add ‘instant book’ option to their listings’ booking channel.
  - If this change was informed to guests as well, thinking of hosts would be more likely to reply, guests would be either less serious in writing their first message or more intended in getting involved in interactions.

- **What is the size of the potential business impact?**
  - Auto-rejection has an impact on decreasing potential bookings.
  - Some hosts might drop some of their listings because of the fact that he/she does not want to undergo the pressure of replying in time meantime losing too many bookings due to the auto-rejection.

Experiment design

- **What are the subjects of the experiment and how will you randomize them into control and treatment?**
  - Subject: individual inquiry event but not host user
  - Since a user may have his/her tendency of replying quickly or slowing. And each user may have several inquiries for the same listing. Having them alway assigned to the same group will introduce bias coming from hosts themselves.
  - Also need to consider keeping the irrelevant metrics being invariant. For example, the boxplot below shows the differences of sample size and behavior regarding replying quickness based on the languages hosts speak. Therefore stratified sampling should be applied.



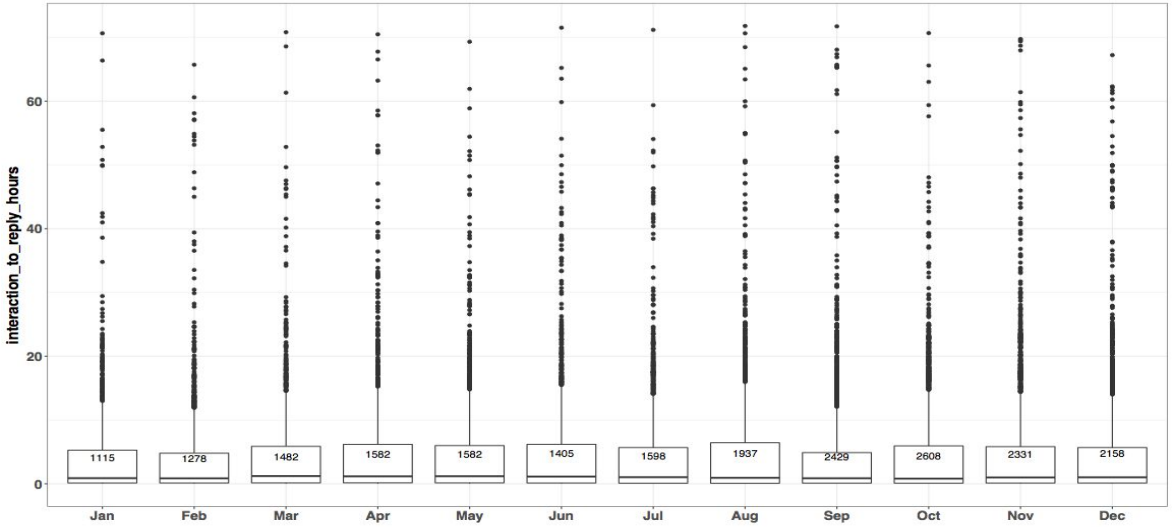
Response time in hours broken down by language
(only samples < 72 hrs shown)

Experiment design

- **Which key metrics will you monitor?**
  - Booking rate
  - Average interaction count per inquiry (since this change could lead to the change of the willingness of guest and host messaging each other)
- **How will decide when to stop the experiment?**
  - Need to factor in seasonal and holiday effect. Even though the historical data does not show that response time much affected by busy/off season (see the plot on next slide), the booking rate or interaction count per inquiry may not.
  - The experiment should be conducted for at least one year.

Experiment design

Response time in hours broken down by month
(only samples < 72 hrs shown)



- **How does this experiment design account for good user experience of the Airbnb product and spillover effects?**
  - It may increase the number of logins/page-views/reviews/host-guest interactions.

Appendix: code

Airbnb Data Science Inference Data Challenge

```
library(data.table)
library(ggplot2)
library(MASS)

folder <- './'
assignmentDT <- data.table::fread(file.path(folder, 'assignments__288_29.csv'))
contactDT <- data.table::fread(file.path(folder, 'contacts__285_29.csv'))
```

Data Sanity Check

```
# Convert variable data types
contactDT[, `:=`(ts_interaction_first = as.POSIXct(ts_interaction_first, format='%Y-%m-%d %H:%M:%S.0', tz
='UTC')
                 , ts_reply_at_first = as.POSIXct(ts_reply_at_first, format='%Y-%m-%d %H:%M:%S.0', tz='UT
C')
                 , ts_accepted_at_first = as.POSIXct(ts_accepted_at_first, format='%Y-%m-%d %H:%M:%S.0',
 tz='UTC')
                 , ts_booking_at = as.POSIXct(ts_booking_at, format='%Y-%m-%d %H:%M:%S.0', tz='UTC')
                 , ds_checkin_first = as.Date(ds_checkin_first)
                 , ds_checkout_first = as.Date(ds_checkout_first)
                 , dim_contact_channel_first = as.factor(dim_contact_channel_first)
                 , dim_room_type = as.factor(dim_room_type)
                 , dim_guest_language = NULL
                 , dim_host_language = as.factor(dim_host_language))]

# summary(contactDT)

# Remove duplicated assignments
assignmentDT[, .(uniq_cnt = length(unique(id_user_anon))), by = ab]
```

```
##           ab uniq_cnt
## 1: treatment     9503
## 2:   control     9566
```

```
assignmentDT <- unique(assignmentDT) # 19069

# Any users assigned to both groups?
assignmentDT[,  .N, by = id_user_anon][N==2, .N]
```

```
## [1] 890
```

```
# Are all the users for testing in the contact data?
all(assignmentDT$id_user_anon %in% unique(contactDT$id_guest_anon))
```

```
## [1] TRUE
```

```
# number of unique guests and hosts in contact data
contactDT[, lapply(.SD, function(x) length(unique(x))), .SDcols = c("id_guest_anon", "id_host_anon")]
```

```
##    id_guest_anon id_host_anon
## 1:         18179         2118
```

```
# Any IDs are both guests and hosts?
length(intersect(unique(contactDT$id_guest_anon), unique(contactDT$id_host_anon)))
```

```
## [1] 28
```

```
# date ranges
contactDT[, lapply(.SD, range, na.rm=TRUE), .SDcols = c("ts_interaction_first"
                                                        , "ts_reply_at_first"
                                                        , "ts_accepted_at_first"
                                                        , "ts_booking_at"
                                                        , "ds_checkin_first"
                                                        , "ds_checkout_first")]
```

```
##    ts_interaction_first   ts_reply_at_first ts_accepted_at_first
## 1:  2013-01-01 02:37:31 2013-01-01 06:05:57  2013-01-01 06:05:57
## 2:  2013-12-31 21:26:22 2016-07-21 12:46:41  2015-07-12 22:07:26
##          ts_booking_at ds_checkin_first ds_checkout_first
## 1: 2013-01-01 06:05:57       2012-12-13        2012-12-17
## 2: 2015-07-12 22:07:26       2016-10-14        2016-10-15
```

```
# Any records with checkin date < interaction_first date | booking_at date?
contactDT[ds_checkin_first < as.Date(ts_interaction_first) | ds_checkin_first < as.Date(ts_booking_at), .
N]
```

```
## [1] 37
```

```
# Any accepted-booking mismatch?
contactDT[!is.na(ts_accepted_at_first) & is.na(ts_booking_at) | is.na(ts_accepted_at_first) & !is.na(ts_b
ooking_at), .N]
```

```
## [1] 54
```

```
contactDT[!is.na(ts_accepted_at_first) & !is.na(ts_booking_at) & ts_accepted_at_first != ts_booking_at, .
N]
```

```
## [1] 102
```

```
# Any instant_booked records not having a booking timestamp?
contactDT[dim_contact_channel_first=='instant_booked', sum(is.na(ts_booking_at))]
```

```
## [1] 0
```

```
# Any the book_it records without an inquiry?
contactDT[dim_contact_channel_first=='book_it', sum(is.na(ts_interaction_first))]
```

```
## [1] 0
```

```
# Any listings having one-to-many mapping to room types?
contactDT[, .(num_room_types = length(unique(dim_room_type))), by = id_listing_anon][num_room_types > 1,
 .N]
```

```
## [1] 0
```

```
# distribution of number of records made by the same guests
contactDT[,  .(guest_record_cnt = .N), by = id_guest_anon][, .N,  by = guest_record_cnt][order(-N)]
```

```
##     guest_record_cnt     N
##  1:                1 14733
##  2:                2  2519
##  3:                3   603
##  4:                4   199
##  5:                5    66
##  6:                6    29
##  7:                7    11
##  8:                8     7
##  9:                9     3
## 10:               10     3
## 11:               11     2
## 12:               12     2
## 13:               13     1
## 14:               14     1
```

```
# How many book_it records not having an acceptance timestamp?
contactDT[dim_contact_channel_first=='book_it', sum(is.na(ts_accepted_at_first))]
```

```
## [1] 6398
```

```
# How many book_it records not recieving a reply?
contactDT[dim_contact_channel_first=='book_it', sum(is.na(ts_reply_at_first))]
```

```
## [1] 1392
```

Q1: Identifying factors that affect booking rate

Initial Model

```
# Add derived input variables: m_stay_days, m_interation_to_checkin_days, m_interaction_to_reply_hours
# and target binary variable: booking
contactDT[, `:=`(m_stay_days = as.integer(ds_checkout_first - ds_checkin_first)
                 , m_interaction_to_checkin_days = as.integer(ds_checkout_first - as.Date(ts_interaction_
first))
                 , m_interaction_to_reply_hours = as.numeric(difftime(ts_reply_at_first, ts_interaction_f
irst, units = "hours"))
                 , booking = ifelse(is.na(ts_booking_at), FALSE, TRUE))]

# logistic regression model 1
bookingProbModel1 <- glm(booking ~ m_guests_first +
                         m_interactions +
                         m_first_message_length_in_characters +
                         dim_room_type +
                         dim_total_reviews +
                         dim_person_capacity +
                         dim_host_language +
                         m_stay_days +
                         m_interaction_to_checkin_days +
                         m_interaction_to_reply_hours
                        , data = contactDT[dim_contact_channel_first=='book_it'], family = binomial(link
=logit))

summary(bookingProbModel1)
```

```
##
## Call:
## glm(formula = booking ~ m_guests_first + m_interactions + m_first_message_length_in_characters +
##     dim_room_type + dim_total_reviews + dim_person_capacity +
##     dim_host_language + m_stay_days + m_interaction_to_checkin_days +
##     m_interaction_to_reply_hours, family = binomial(link = logit),
##     data = contactDT[dim_contact_channel_first == "book_it"])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3731  -0.7534   0.3212   0.7087   8.4904
##
## Coefficients:
##                                      Estimate Std. Error z value
## (Intercept)                         -1.1460637  0.3186816  -3.596
## m_guests_first                       0.0262324  0.0167794   1.563
## m_interactions                       0.3636278  0.0084549  43.008
## m_first_message_length_in_characters -0.0005744  0.0001129  -5.086
## dim_room_typePrivate room           -0.5611509  0.0458756 -12.232
## dim_room_typeShared room             0.0134750  0.1763985   0.076
## dim_total_reviews                    0.0156710  0.0006142  25.513
## dim_person_capacity                 -0.1090111  0.0095264 -11.443
## dim_host_languageen                  0.0805961  0.3203500   0.252
## dim_host_languagees                  0.0637729  0.3119962   0.204
## dim_host_languagefr                 -0.3521083  0.3736043  -0.942
## dim_host_languageit                  0.0611943  0.5399081   0.113
## dim_host_languagenl                 -0.4627842  1.2599450  -0.367
## dim_host_languagept                  0.4871799  0.8412704   0.579
## m_stay_days                         -0.0044715  0.0024948  -1.792
## m_interaction_to_checkin_days        0.0016395  0.0005684   2.885
## m_interaction_to_reply_hours        -0.0085199  0.0011122  -7.660
##                                     Pr(>|z|)
## (Intercept)                         0.000323 ***
## m_guests_first                      0.117966
## m_interactions                       < 2e-16 ***
## m_first_message_length_in_characters 3.65e-07 ***
## dim_room_typePrivate room            < 2e-16 ***
## dim_room_typeShared room            0.939109
## dim_total_reviews                    < 2e-16 ***
## dim_person_capacity                  < 2e-16 ***
## dim_host_languageen                 0.801360
## dim_host_languagees                 0.838039
## dim_host_languagefr                 0.345955
## dim_host_languageit                 0.909759
## dim_host_languagenl                 0.713391
## dim_host_languagept                 0.562522
## m_stay_days                         0.073080 .
## m_interaction_to_checkin_days       0.003920 **
## m_interaction_to_reply_hours        1.85e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 19681  on 16784  degrees of freedom
## Residual deviance: 14377  on 16768  degrees of freedom
##   (3073 observations deleted due to missingness)
## AIC: 14411
##
## Number of Fisher Scoring iterations: 8
```

Model Selection

```
# model selection using stepwise AIC
MASS::stepAIC(bookingProbModel1)
```

```
## Start:  AIC=14410.62
## booking ~ m_guests_first + m_interactions + m_first_message_length_in_characters +
##     dim_room_type + dim_total_reviews + dim_person_capacity +
##     dim_host_language + m_stay_days + m_interaction_to_checkin_days +
##     m_interaction_to_reply_hours
##
##                                       Df Deviance   AIC
## - dim_host_language                    6    14381 14403
## <none>                                      14377 14411
## - m_guests_first                       1    14379 14411
## - m_stay_days                          1    14379 14411
## - m_interaction_to_checkin_days        1    14389 14421
## - m_first_message_length_in_characters 1    14400 14432
## - dim_person_capacity                  1    14506 14538
## - dim_room_type                        2    14522 14552
## - m_interaction_to_reply_hours         1    14540 14572
## - dim_total_reviews                    1    15104 15136
## - m_interactions                       1    17570 17602
##
## Step:  AIC=14402.67
## booking ~ m_guests_first + m_interactions + m_first_message_length_in_characters +
##     dim_room_type + dim_total_reviews + dim_person_capacity +
##     m_stay_days + m_interaction_to_checkin_days + m_interaction_to_reply_hours
##
##                                       Df Deviance   AIC
## <none>                                      14381 14403
## - m_guests_first                       1    14383 14403
## - m_stay_days                          1    14383 14403
## - m_interaction_to_checkin_days        1    14393 14413
## - m_first_message_length_in_characters 1    14404 14424
## - dim_person_capacity                  1    14511 14531
## - dim_room_type                        2    14526 14544
## - m_interaction_to_reply_hours         1    14545 14565
## - dim_total_reviews                    1    15114 15134
## - m_interactions                       1    17573 17593
```

```
##
## Call:  glm(formula = booking ~ m_guests_first + m_interactions + m_first_message_length_in_characters
+
##     dim_room_type + dim_total_reviews + dim_person_capacity +
##     m_stay_days + m_interaction_to_checkin_days + m_interaction_to_reply_hours,
##     family = binomial(link = logit), data = contactDT[dim_contact_channel_first ==
##         "book_it"])
##
## Coefficients:
##                           (Intercept)
##                             -1.083999
##                         m_guests_first
##                              0.026206
##                         m_interactions
##                              0.363410
## m_first_message_length_in_characters
##                             -0.000575
##             dim_room_typePrivate room
##                             -0.558549
##              dim_room_typeShared room
##                              0.023960
##                       dim_total_reviews
##                              0.015655
##                    dim_person_capacity
##                             -0.109220
##                            m_stay_days
##                             -0.004445
##         m_interaction_to_checkin_days
##                              0.001656
##          m_interaction_to_reply_hours
##                             -0.008551
##
## Degrees of Freedom: 16784 Total (i.e. Null);  16774 Residual
##   (3073 observations deleted due to missingness)
## Null Deviance:       19680
## Residual Deviance: 14380     AIC: 14400
```

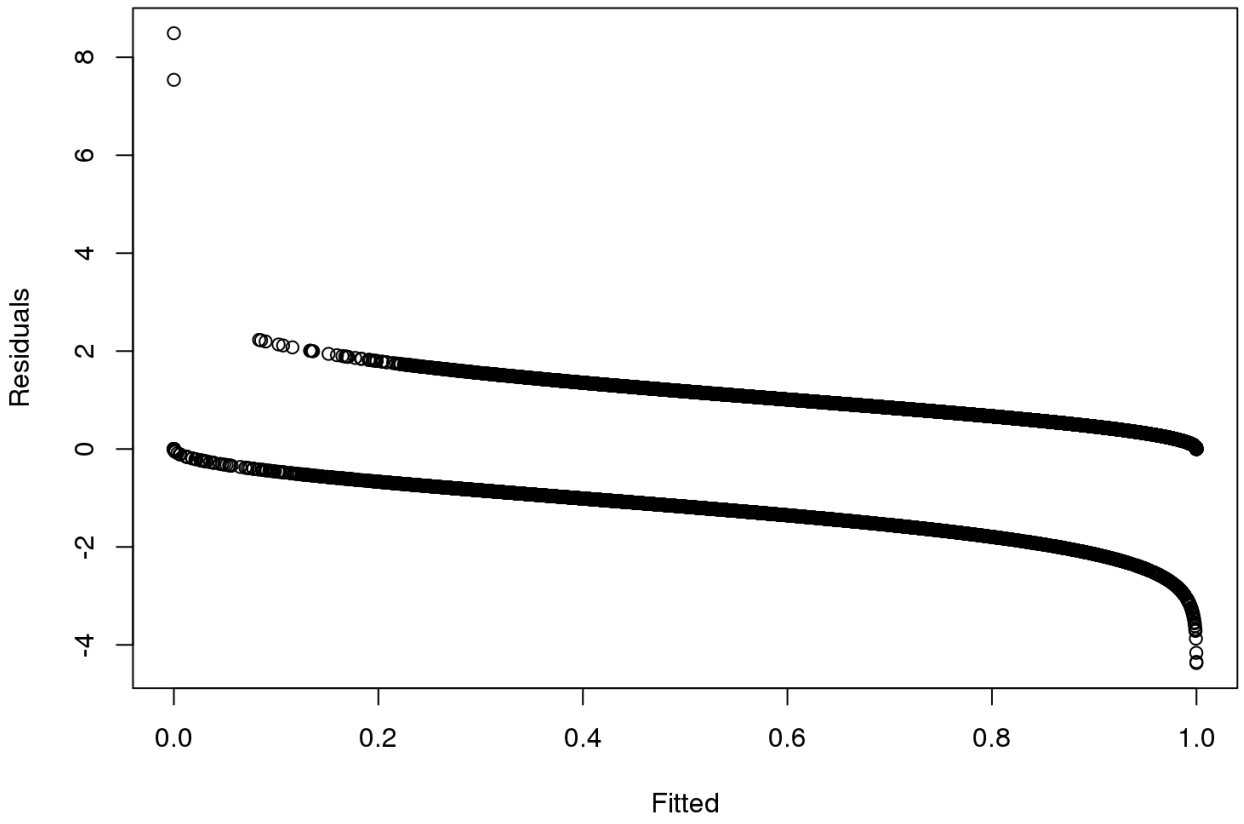
Updated Model

```
# logistic regression model 2
# remove host_language
bookingProbModel2 <- glm(booking ~ m_guests_first +
                           m_interactions +
                           m_first_message_length_in_characters +
                           dim_room_type +
                           dim_total_reviews +
                           dim_person_capacity +
                           m_stay_days +
                           m_interaction_to_checkin_days +
                           m_interaction_to_reply_hours
                         , data = contactDT[dim_contact_channel_first=='book_it'], family = binomial(link
=logit))

summary(bookingProbModel2)
```

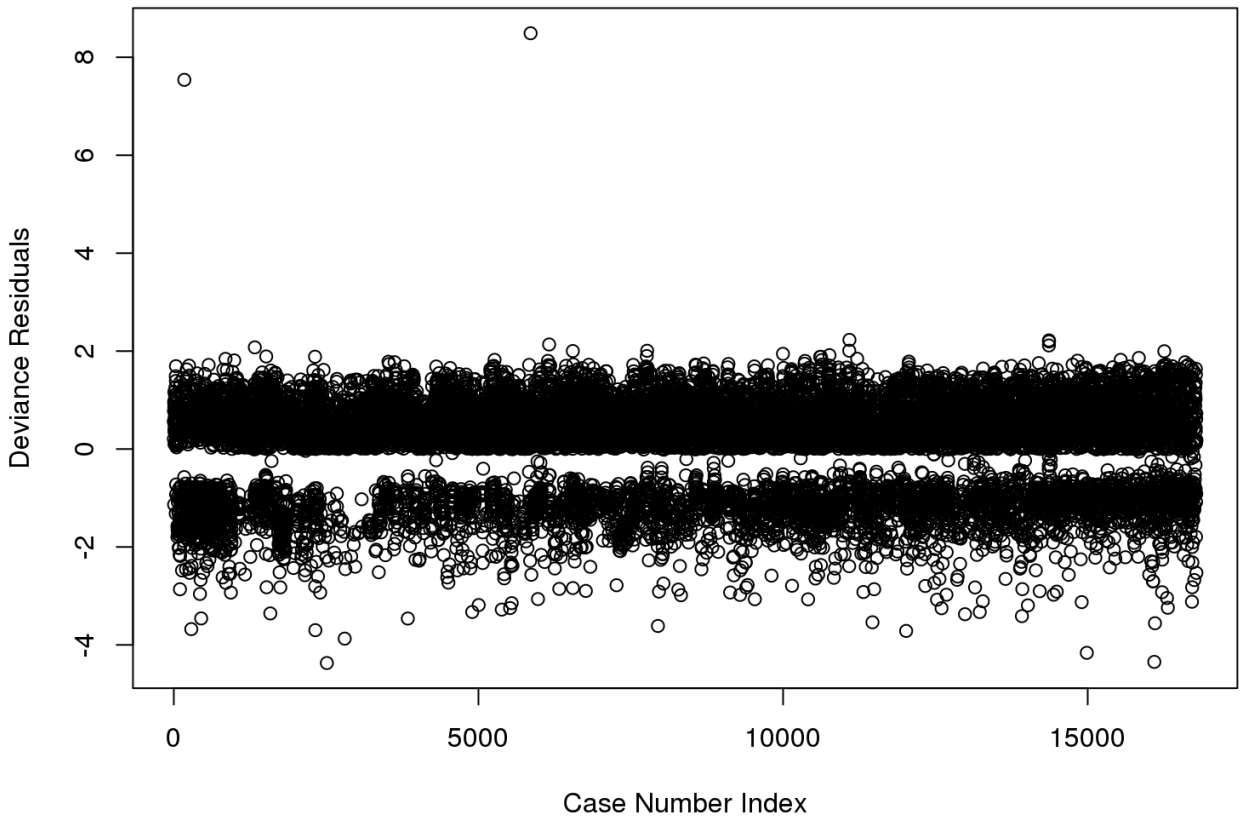
```
##
## Call:
## glm(formula = booking ~ m_guests_first + m_interactions + m_first_message_length_in_characters +
##     dim_room_type + dim_total_reviews + dim_person_capacity +
##     m_stay_days + m_interaction_to_checkin_days + m_interaction_to_reply_hours,
##     family = binomial(link = logit), data = contactDT[dim_contact_channel_first ==
##         "book_it"])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3704  -0.7541   0.3216   0.7098   8.4904
##
## Coefficients:
##                                     Estimate Std. Error z value
## (Intercept)                       -1.0839989  0.0743507 -14.580
## m_guests_first                     0.0262058  0.0167756   1.562
## m_interactions                     0.3634100  0.0084501  43.006
## m_first_message_length_in_characters -0.0005750  0.0001129  -5.095
## dim_room_typePrivate room         -0.5585493  0.0458001 -12.195
## dim_room_typeShared room           0.0239599  0.1748407   0.137
## dim_total_reviews                  0.0156546  0.0006100  25.665
## dim_person_capacity               -0.1092197  0.0095190 -11.474
## m_stay_days                       -0.0044452  0.0024946  -1.782
## m_interaction_to_checkin_days      0.0016561  0.0005684   2.914
## m_interaction_to_reply_hours      -0.0085507  0.0011096  -7.706
##                                   Pr(>|z|)
## (Intercept)                        < 2e-16 ***
## m_guests_first                     0.11826
## m_interactions                     < 2e-16 ***
## m_first_message_length_in_characters 3.48e-07 ***
## dim_room_typePrivate room          < 2e-16 ***
## dim_room_typeShared room           0.89100
## dim_total_reviews                  < 2e-16 ***
## dim_person_capacity                < 2e-16 ***
## m_stay_days                        0.07476 .
## m_interaction_to_checkin_days      0.00357 **
## m_interaction_to_reply_hours       1.30e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 19681  on 16784  degrees of freedom
## Residual deviance: 14381  on 16774  degrees of freedom
##   (3073 observations deleted due to missingness)
## AIC: 14403
##
## Number of Fisher Scoring iterations: 8
```

Model Diagnostic

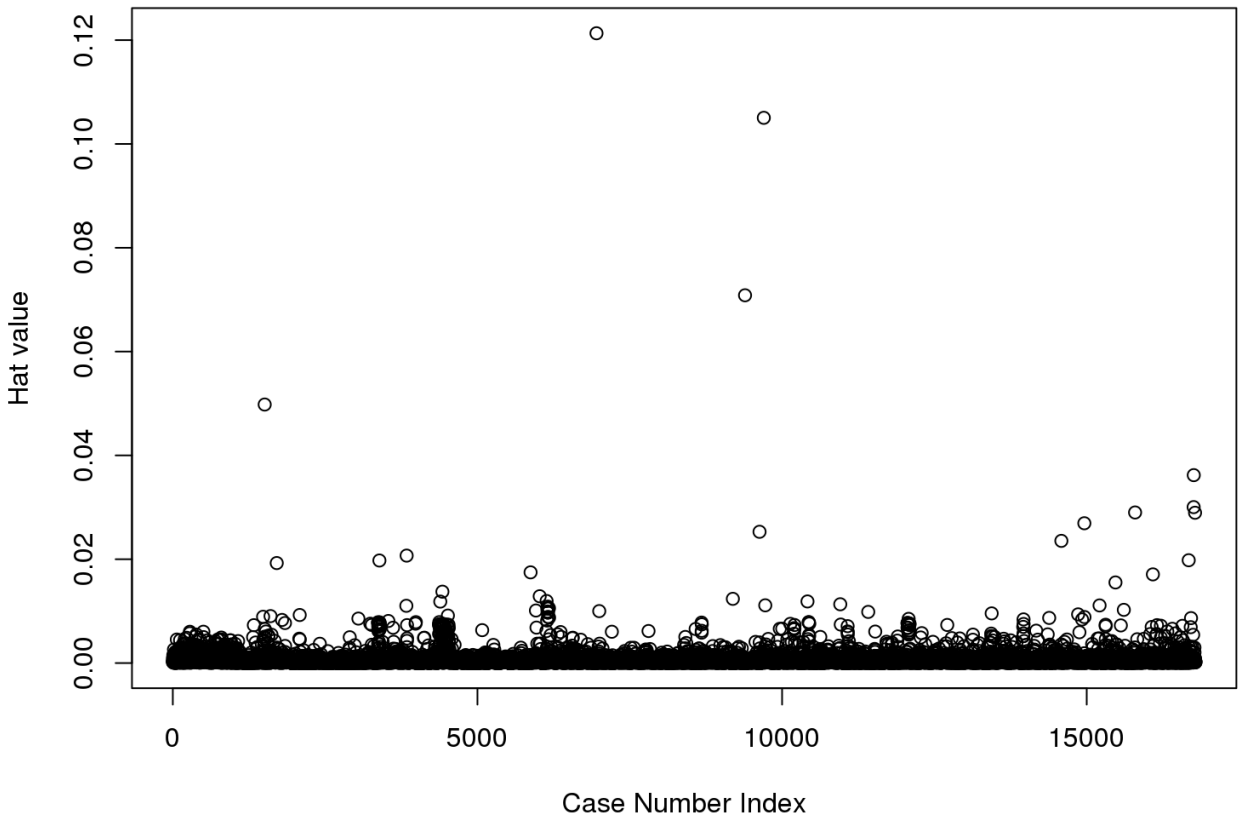
```
plot(fitted(bookingProbModel2), residuals(bookingProbModel2), xlab = 'Fitted', ylab = 'Residuals')
```




```
plot(residuals(bookingProbModel2, type="deviance"), xlab = "Case Number Index", ylab = 'Deviance Residual
s')
```

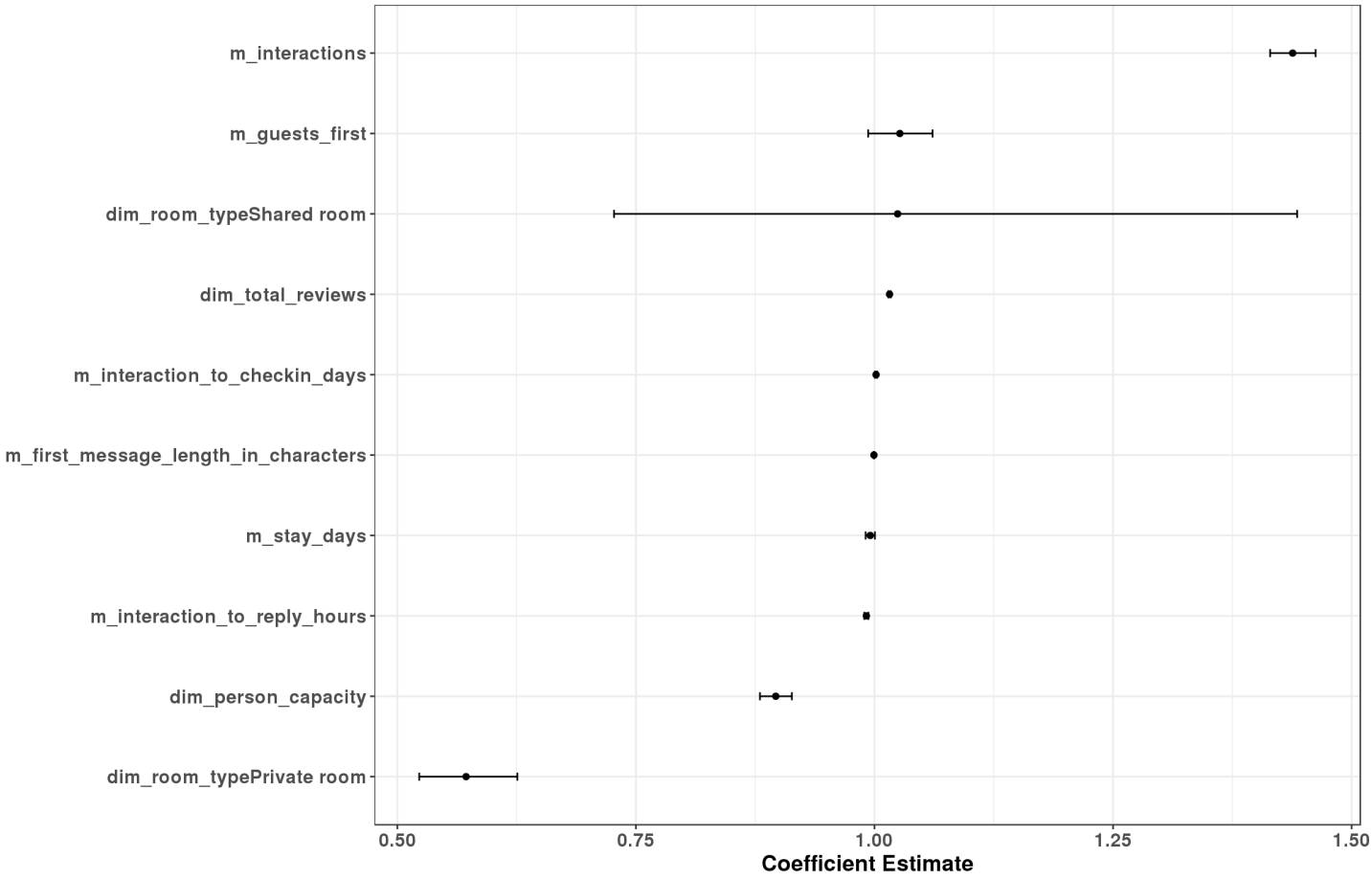



```
plot(hatvalues(bookingProbModel2), ylab="Hat value", xlab="Case Number Index")
```



Model Interpretation

```
# Exponentiate estimated coefficients in order to get odds-ratio interpretations
expCoefs <- exp(coefficients(bookingProbModel2))
# Exponentiate Wald confidence intervals to get on the odds-scale
expCoefCIs <- exp(confint.default(bookingProbModel2, level=0.95))
expCoefsDT <- data.table(data.frame(cbind(expCoefs,expCoefCIs)), keep.rownames=TRUE)
colnames(expCoefsDT) <- c("variable", "point_est", "CI_lower", "CI_upper")
expCoefsDT <- expCoefsDT[variable!='(Intercept)'][order(-point_est)]

# plot ranked (exponentiated) coefficent estimates and CIs
MakeConfIntPlot <- function(DT, xPt, xLower, xUpper, yVar, groupVar=NULL, xLab='', title='', ...){
  p <- ggplot2::ggplot(data = DT, aes_string(x=xPt, y=yVar, color=groupVar)) +
    ggplot2::geom_point() +
    ggplot2::geom_errorbarh(aes_string(xmax = xUpper, xmin = xLower, height = .1)) +
    ggplot2::theme_bw() +
    ggplot2::labs(x = xLab, y = NULL) +
    ggplot2::ggtitle(title) +
    ggplot2::theme(axis.text=element_text(size=12, face="bold"),
                   axis.title=element_text(size=14, face="bold"),
                   legend.text=element_text(size=12),
                   legend.title=element_text(size=12),
                   title=element_text(size=16, face="bold"))
  return(p)
}

expCoefsDT[, variable := factor(variable, levels=expCoefsDT[order(point_est),variable])]
p <- MakeConfIntPlot(expCoefsDT, xPt = 'point_est', xLower = 'CI_lower', xUpper = 'CI_upper', yVar = 'var
iable', xLab='Coefficient Estimate')
print(p)
```



Q2: A/B testing on effects of minimum word count requirement

```
# Remove bias introduced by including missing value cases only in control group
contactDT[, sum(is.na(m_first_message_length_in_characters)), by=dim_contact_channel_first]
```

```
##    dim_contact_channel_first   V1
## 1:                   book_it 1891
## 2:            instant_booked  335
```

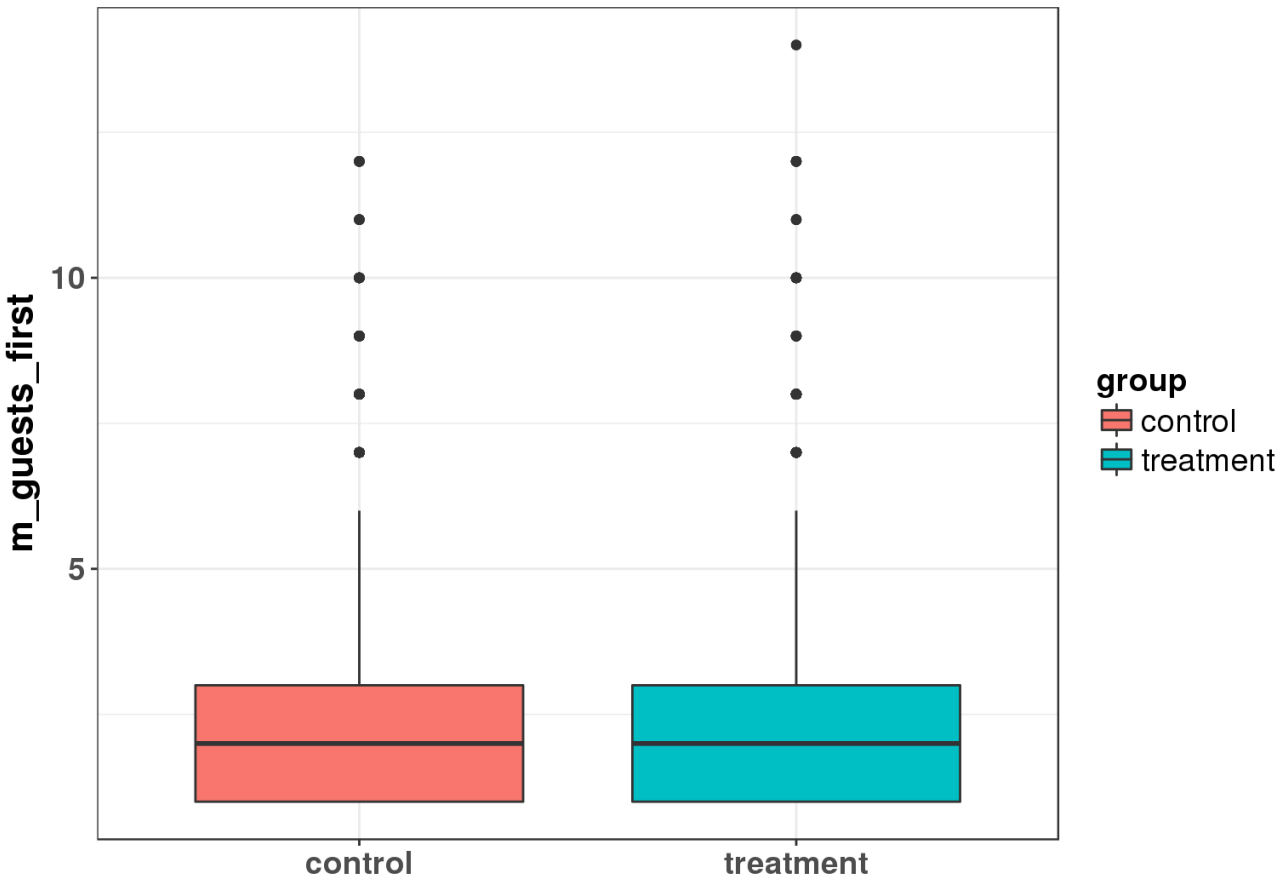
```
contactDT.ab <- contactDT[!is.na(m_first_message_length_in_characters),]

ids_a <- assignmentDT[ab=='treatment', id_user_anon]
ids_b <- assignmentDT[ab=='control', id_user_anon]

# Some subjects in treatment group are not associated with messages having > 140 words
all(ids_a %in% contactDT.ab[m_first_message_length_in_characters >= 140, unique(id_guest_anon)])
```

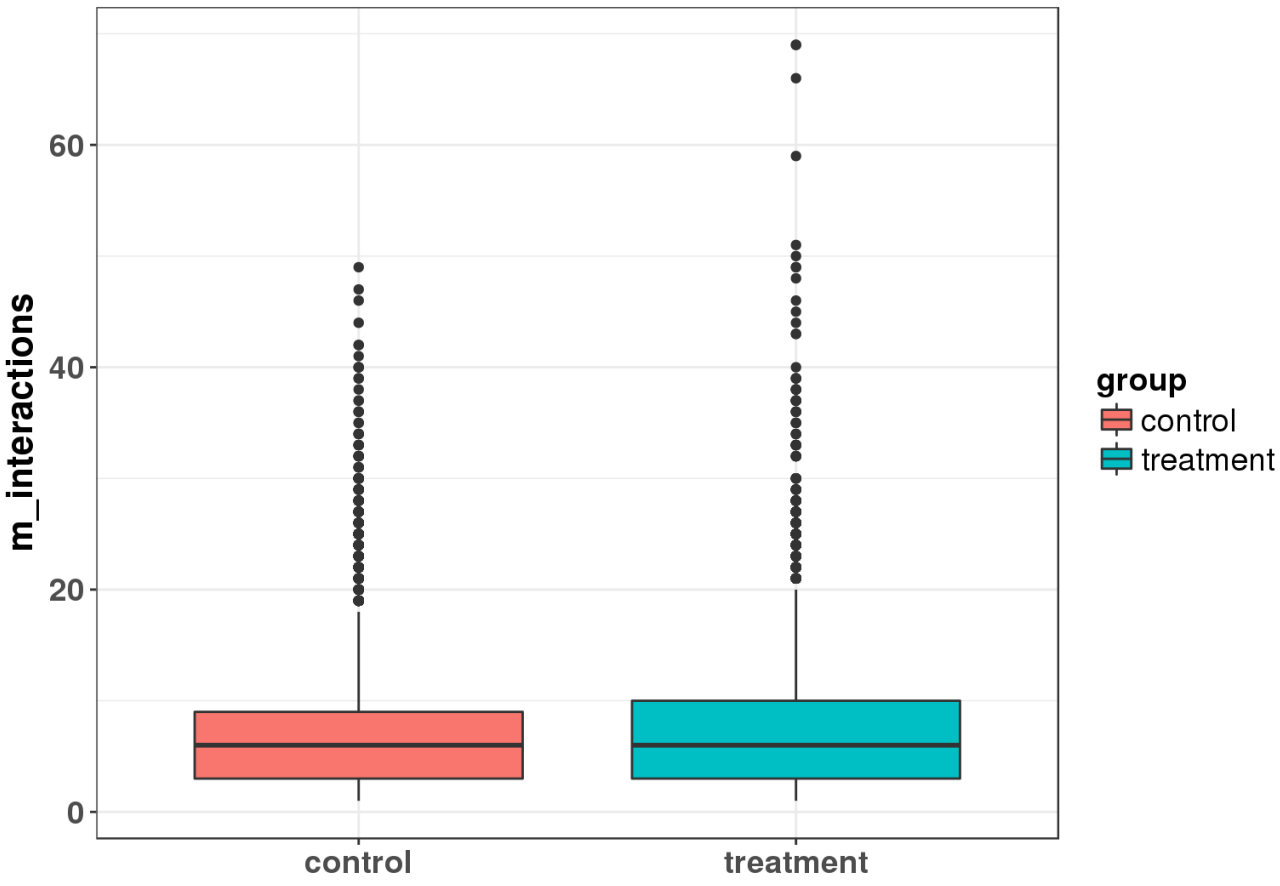
```
## [1] FALSE
```

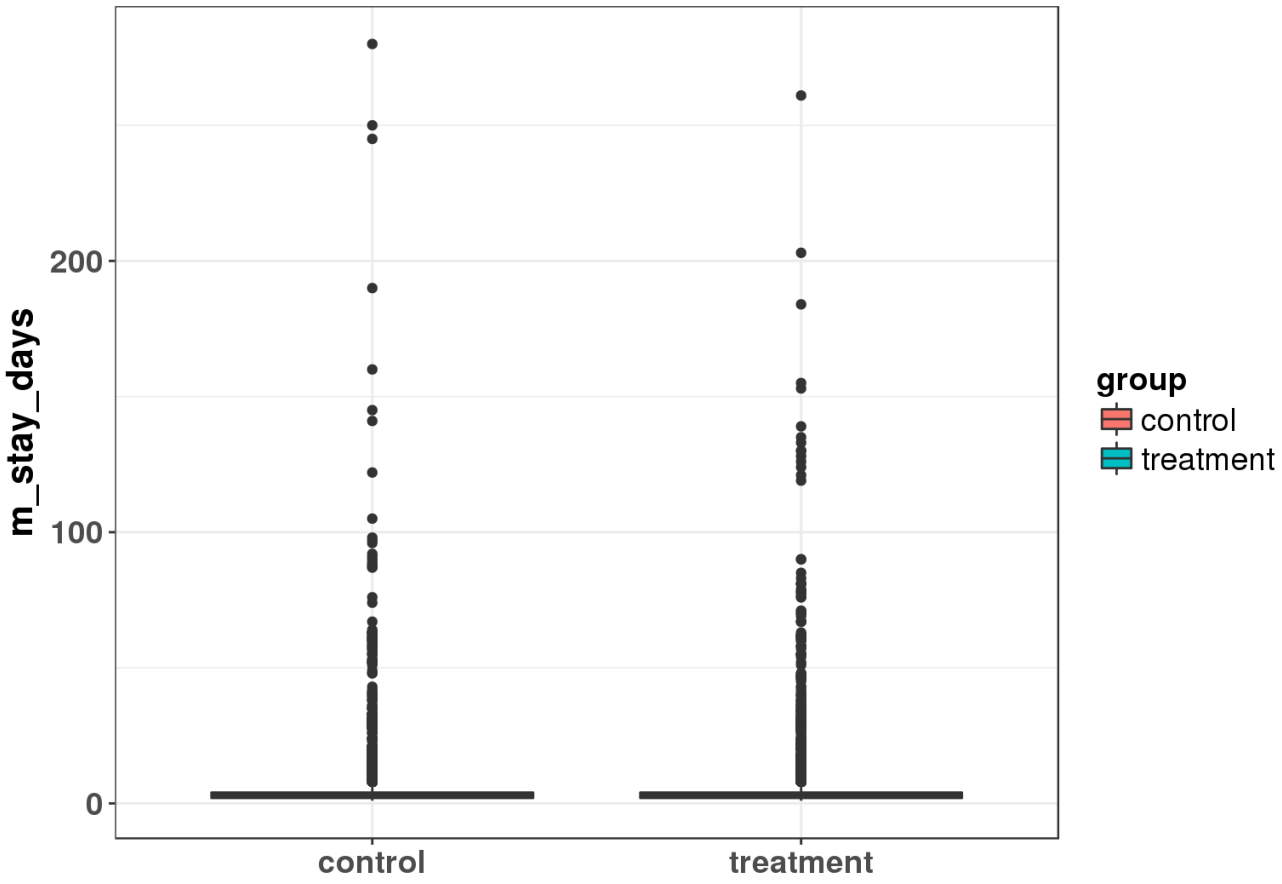
```
contactDT.ab[, group := rep('', .N)]
contactDT.ab[id_guest_anon %in% ids_a & m_first_message_length_in_characters >= 140, group := 'treatment'
]
contactDT.ab[id_guest_anon %in% ids_b & group == '', group := 'control']
contactDT.ab <- contactDT.ab[group!='',][, group := as.factor(group)]
```

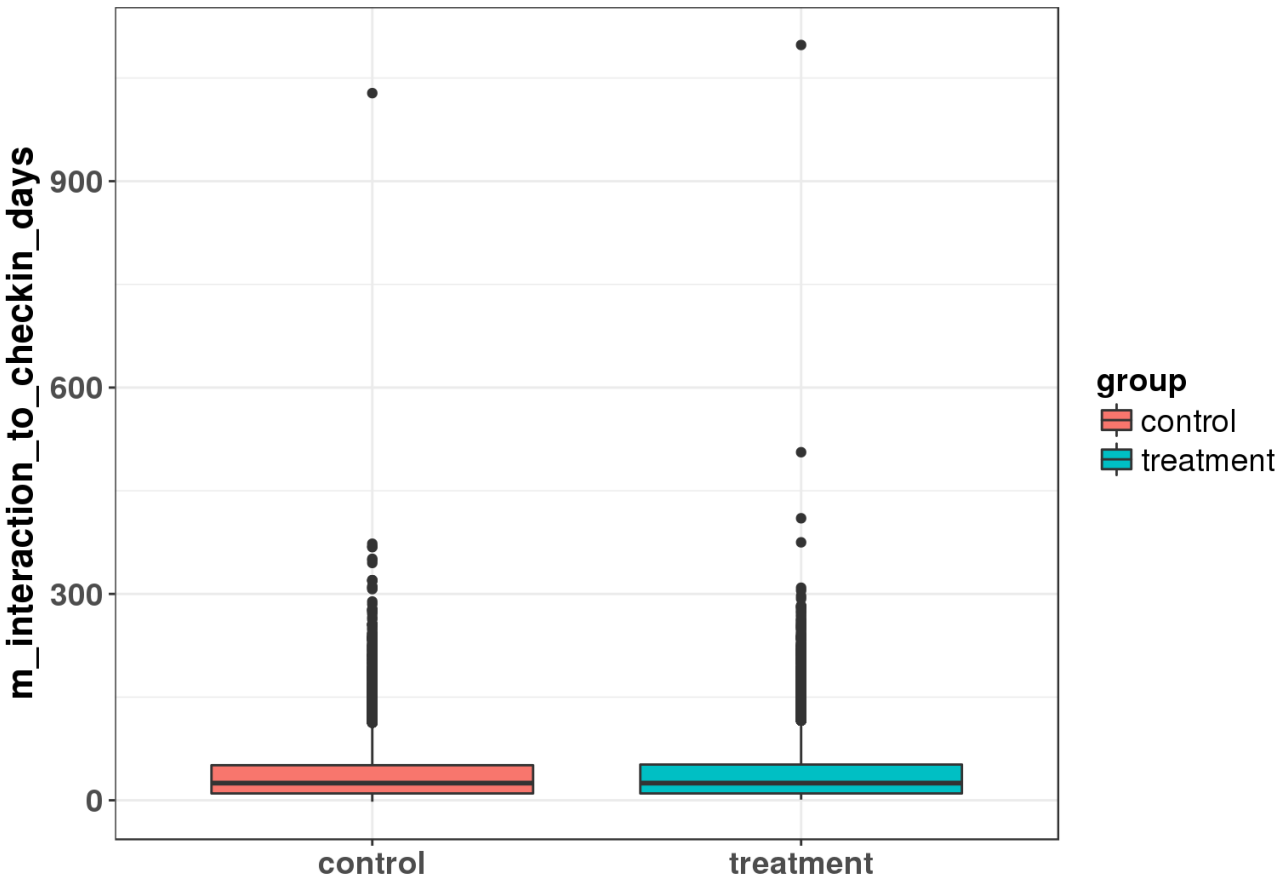
Check invariant metrics over different groups

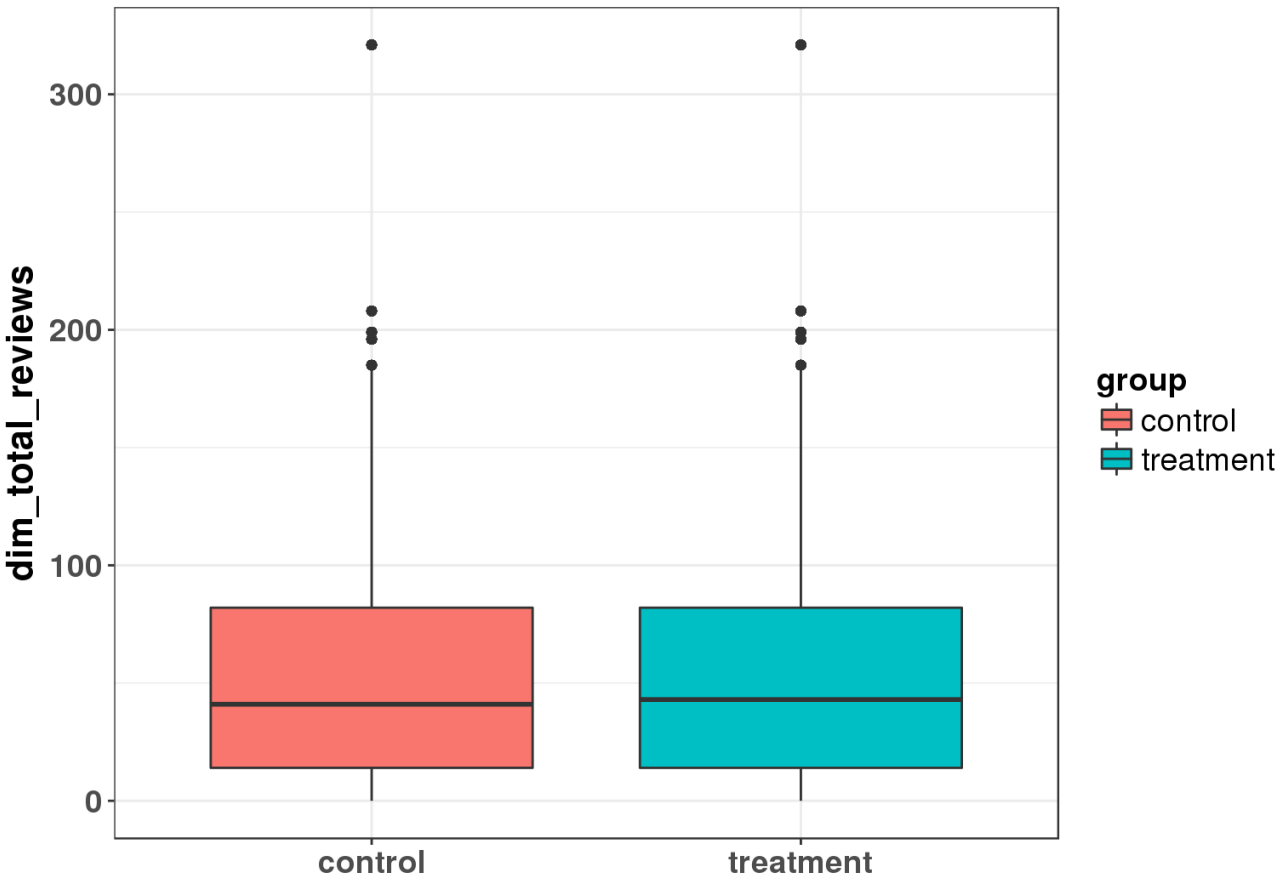
```
# numerical variables
MakeMetricBoxplot <- function(DT, xVar, yVar, groupVar=NULL, xLab='', yLab='', title='', ...){
  p <- ggplot2::ggplot(DT, aes_string(x = xVar, y = yVar, fill = groupVar)) +
    ggplot2::geom_boxplot(...) +
    ggplot2::theme_bw()+
    ggplot2::xlab(xLab) +
    ggplot2::ylab(yLab) +
    ggplot2::ggtitle(title) +
    ggplot2::theme(axis.text=element_text(size=14, face="bold"),
                   axis.title=element_text(size=16, face="bold"),
                   legend.text=element_text(size=14),
                   legend.title=element_text(size=14),
                   title=element_text(size=18, face="bold"))
  return(p)
}

numericVars <- c("m_guests_first", "m_interactions", "m_stay_days", "m_interaction_to_checkin_days", "dim
_total_reviews", "dim_person_capacity")
for(numVar in numericVars){
  p <- MakeMetricBoxplot(contactDT.ab, xVar = 'group', yVar = numVar, groupVar = 'group', yLab = numVar)
  print(p)
}
```

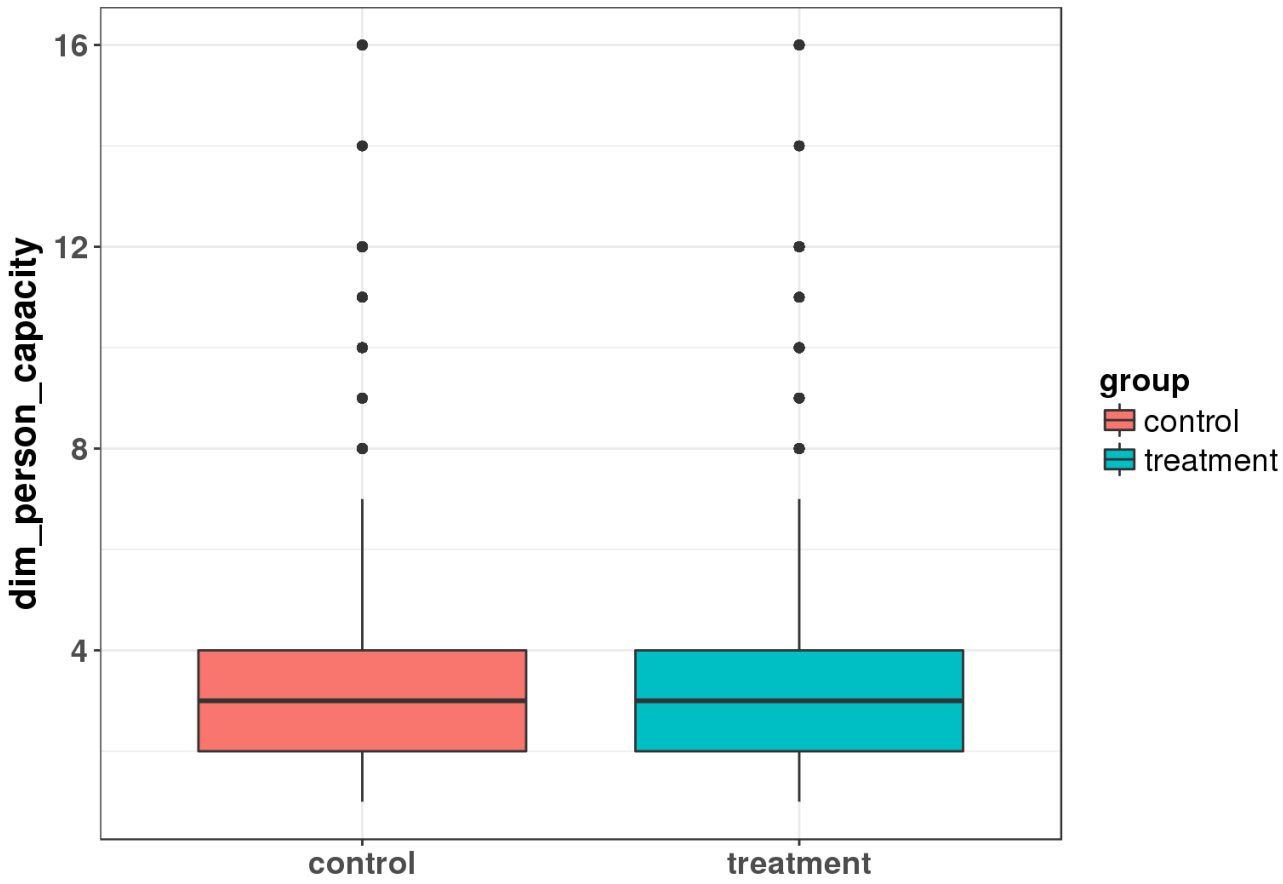

m_guests_first
10
5
control
treatment
group
control
treatment


m_interactions
60
40
20
0
control
treatment
group
control
treatment


m_stay_days
200
100
0
control
treatment
group
control
treatment


m_interaction_to_checkin_days
900
600
300
0
control
treatment
group
control
treatment


dim_total_reviews
300
200
100
0
control
treatment
group
control
treatment




```
# categorical variables
catVars <- c("dim_contact_channel_first", "dim_room_type", "dim_host_language")
for(catVar in catVars){
  print(contactDT.ab[, .N, by=c(catVar, "group")])
}
```

```
##    dim_contact_channel_first     group    N
## 1:                   book_it   control 8728
## 2:                   book_it treatment 8763
## 3:            instant_booked   control 1417
## 4:            instant_booked treatment 1432
##      dim_room_type     group    N
## 1:    Private room   control 3052
## 2:    Private room treatment 3209
## 3: Entire home/apt treatment 6853
## 4: Entire home/apt   control 6959
## 5:     Shared room   control  134
## 6:     Shared room treatment  133
##     dim_host_language     group    N
##  1:                es   control 9200
##  2:                es treatment 9217
##  3:                fr   control  113
##  4:                fr treatment  126
##  5:                de   control   48
##  6:                de treatment   45
##  7:                en   control  758
##  8:                en treatment  777
##  9:                ru treatment    4
## 10:                nl treatment    2
## 11:                it   control   17
## 12:                pt   control    6
## 13:                it treatment   16
## 14:                pt treatment    8
## 15:                nl   control    2
## 16:                da   control    1
```

```
# booking rate two proportion z-test
contactDT.ab[, .(booking_rate = sum(booking)/.N), by=group]
```

```
##        group booking_rate
## 1:   control    0.7298176
## 2: treatment    0.7225110
```

```
prop.test(table(contactDT.ab$group, !contactDT.ab$booking), alternative='two.sided', correct=FALSE)
```

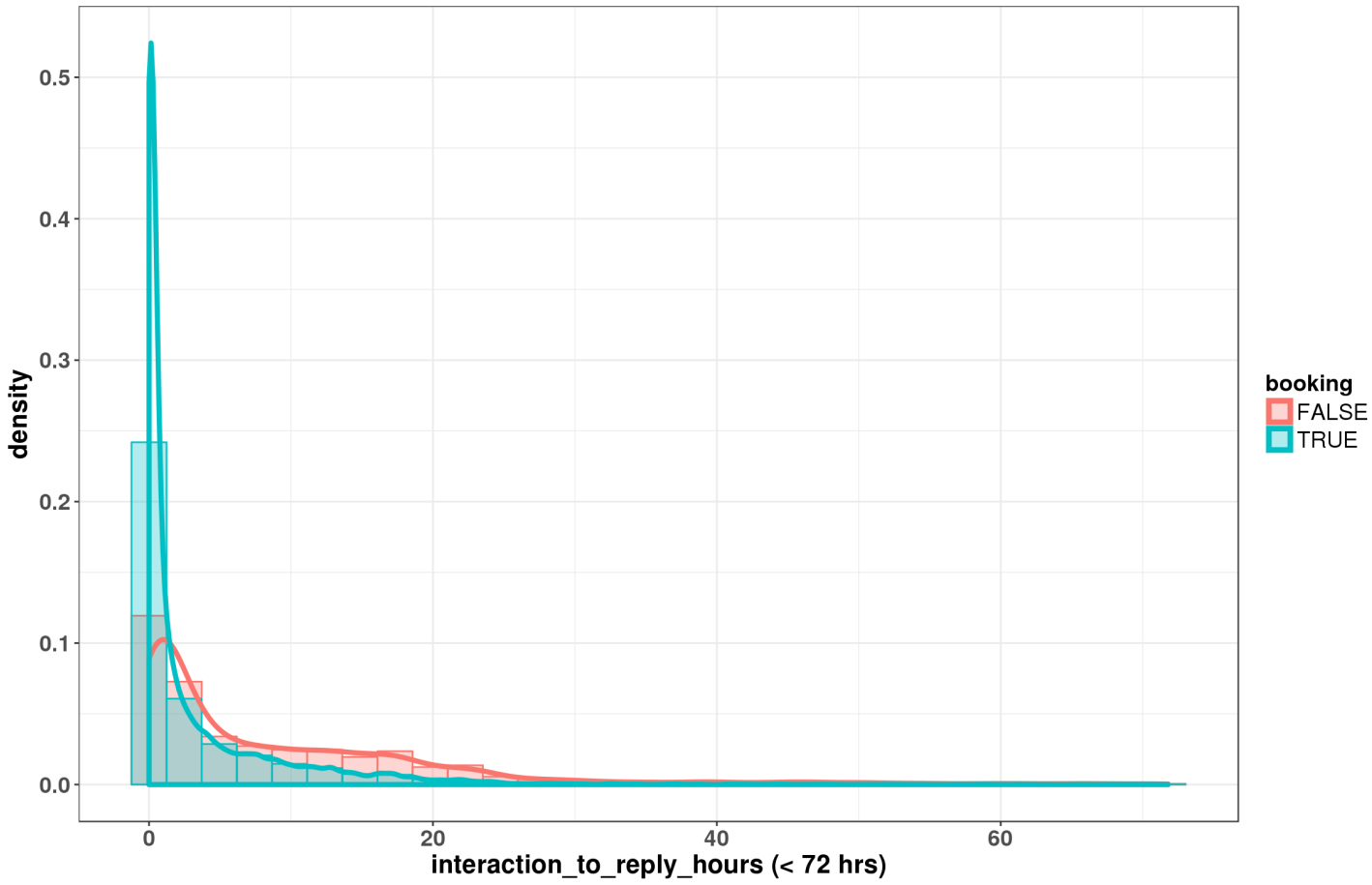
```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  table(contactDT.ab$group, !contactDT.ab$booking)
## X-squared = 1.3652, df = 1, p-value = 0.2426
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.004949352  0.019562570
## sample estimates:
##    prop 1    prop 2
## 0.7298176 0.7225110
```

Q3: Test for change of response time limit

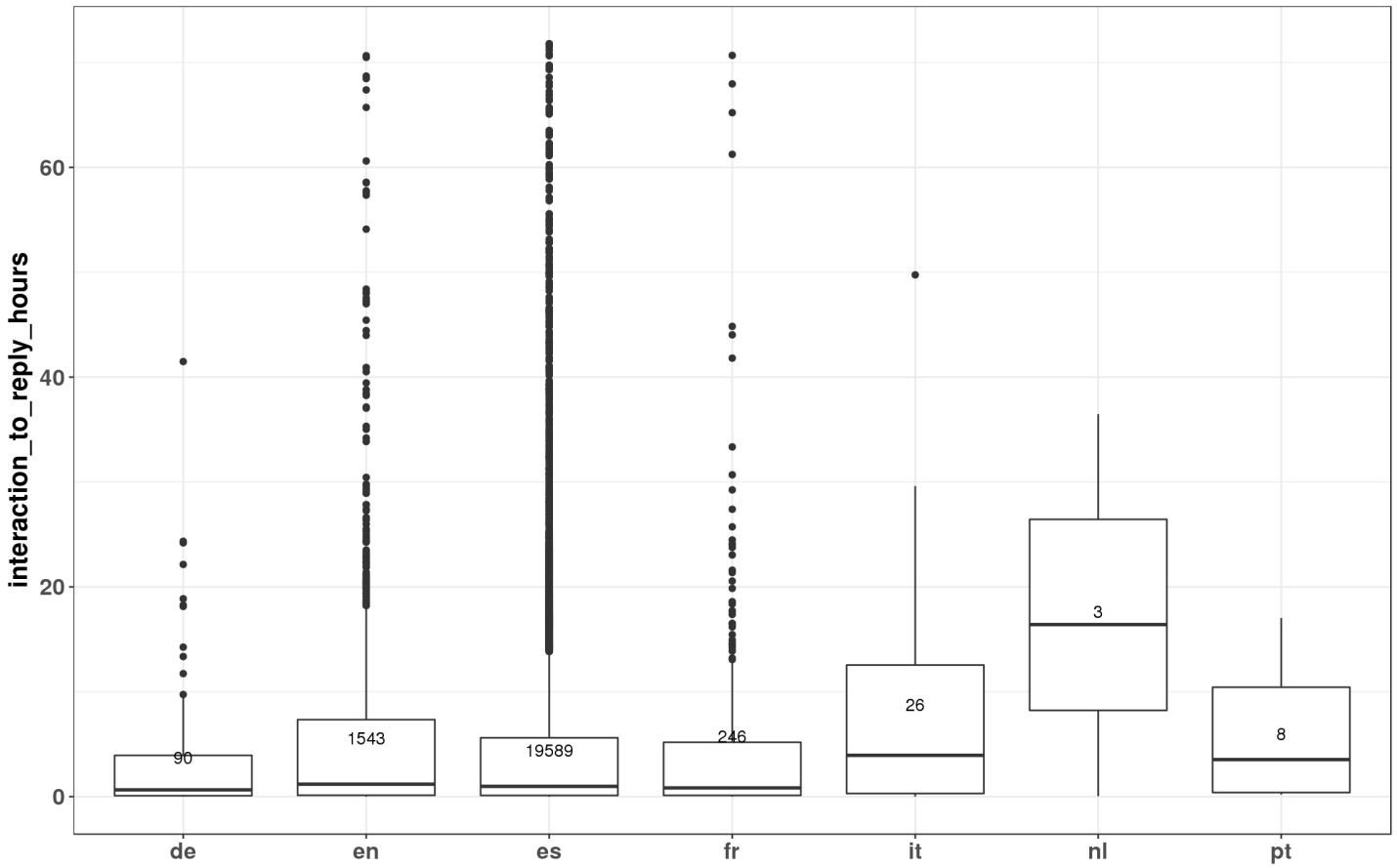
```
# response time quantiles
summary(contactDT$m_interaction_to_reply_hours)
```

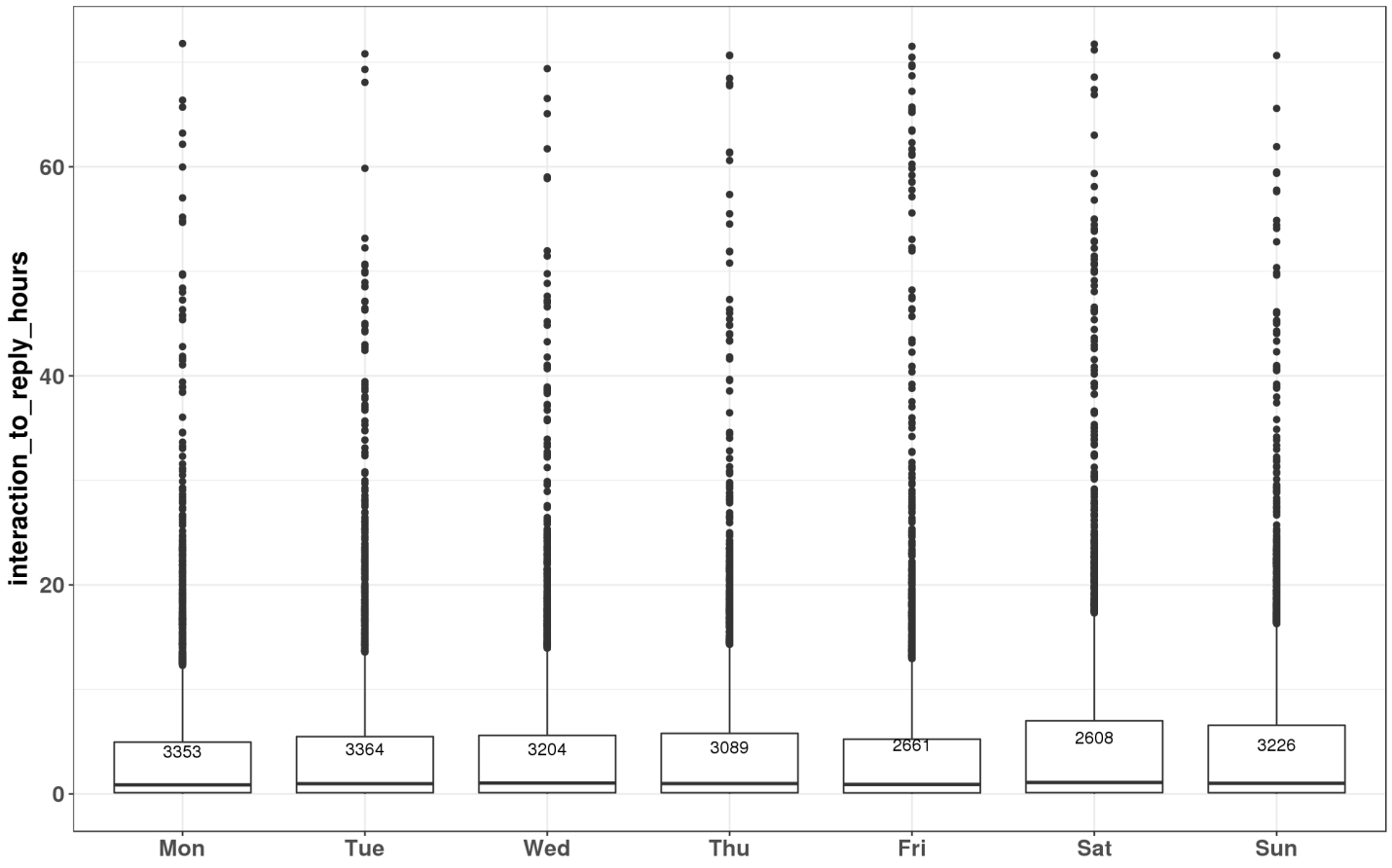
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
##     0.000     0.118     1.030    11.688     6.060 26282.246      1392
```

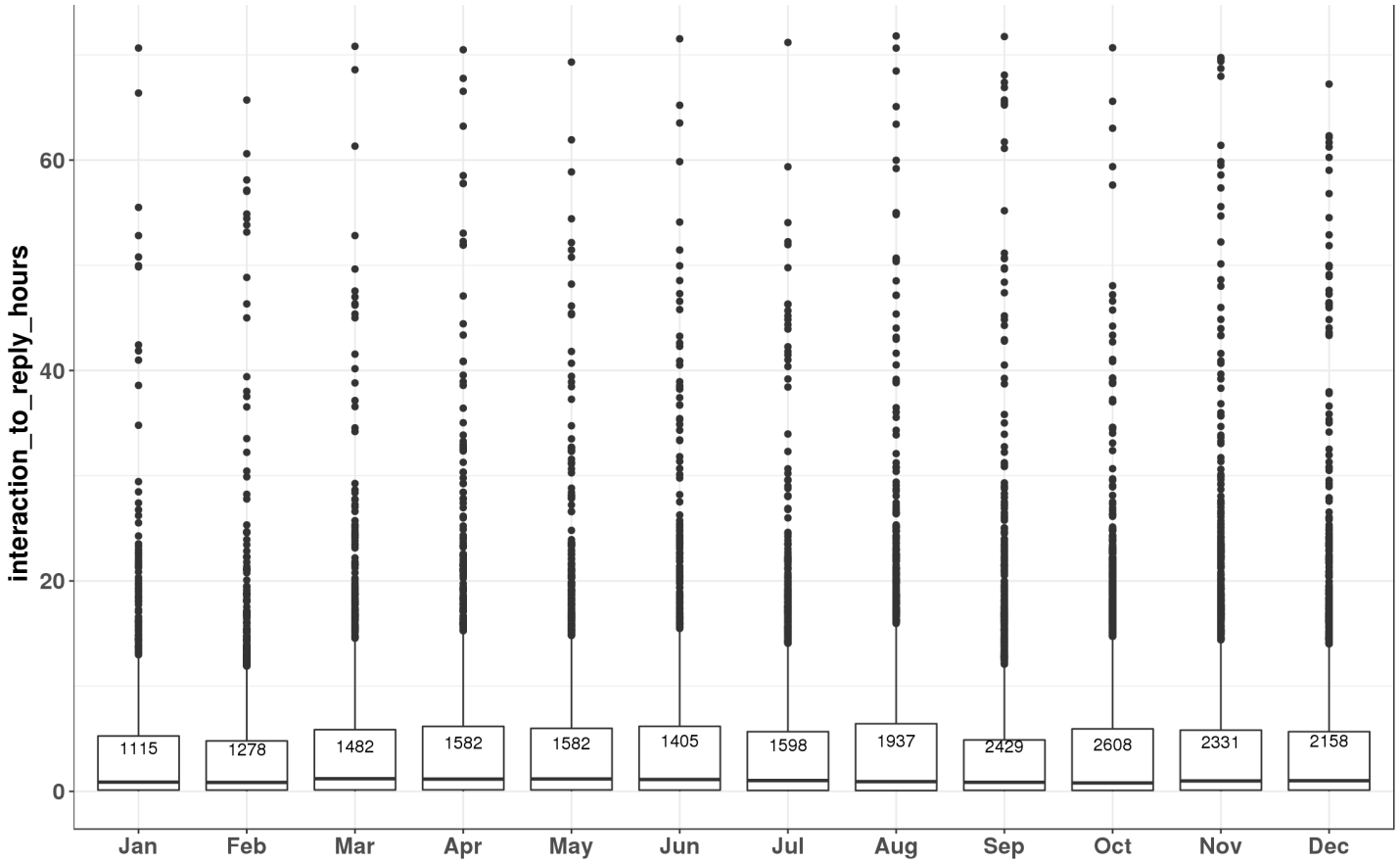
```
# Consider cases with first replies received within 3 days
contactDT.timeTest <- contactDT[m_interaction_to_reply_hours < 3*24]

MakeHistDenPlot <- function(DT, xVar, groupVar=NULL, bins=30, position="identity",  xLab='', title='',
...){
  p <- ggplot2::ggplot(DT, mapping = aes_string(x=xVar, fill=groupVar, color=groupVar), ...) +
    ggplot2::geom_histogram(mapping = aes(y=..density..),
                            bins=bins, position=position, alpha=.3) +
    ggplot2::geom_density(mapping = aes_string(color=groupVar), alpha=0, size=1.5) +
    ggplot2::theme_bw() +
    ggplot2::xlab(xLab) +
    ggplot2::ggtitle(title) +
    ggplot2::theme(axis.text=element_text(size=14, face="bold"),
                   axis.title=element_text(size=16, face="bold"),
                   legend.text=element_text(size=14),
                   legend.title=element_text(size=14),
                   title=element_text(size=18, face="bold"))
  return(p)
}
p <- MakeHistDenPlot(contactDT.timeTest, xVar = 'm_interaction_to_reply_hours', groupVar = 'booking', xLa
b='interaction_to_reply_hours (< 72 hrs)')
print(p)
```



```
contactDT.timeTest[, `:=`(dow = ordered(weekdays(ts_interaction_first, abbreviate=TRUE), levels = c("Mon"
,"Tue","Wed","Thu","Fri","Sat","Sun"))
                      , month = ordered(months(ts_interaction_first, abbreviate=TRUE), levels = c("Jan","F
eb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")))]

groupVars <- c("dim_host_language", "dow", "month")
for(grpVar in groupVars){
  p <- MakeMetricBoxplot(contactDT.timeTest, xVar = grpVar, yVar = 'm_interaction_to_reply_hours'
                         , groupVar = NULL, yLab = 'interaction_to_reply_hours')
  p <- p + stat_summary(fun.data = function(x){c(y = mean(x), label = length(x))}, geom = "text")
  print(p)
}
```


interaction_to_reply_hours
60
40
20
0
90
1543
19589
246
26
3
8
de
en
es
fr
it
nl
pt


interaction_to_reply_hours
60
40
20
0
3353
3364
3204
3089
2661
2608
3226
Mon
Tue
Wed
Thu
Fri
Sat
Sun


interaction_to_reply_hours
60
40
20
0
1115
1278
1482
1582
1582
1405
1598
1937
2429
2608
2331
2158
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec