Exploratory Data Analysis

Gustavo Mello

2022-03-19

Data Loading and Cleansing

Data Loading

```
cwd <- getwd()
listings <- as_tibble(read.csv(paste0(cwd, "/../data/input/listings-challenge.csv")))
daily.revenue <- as_tibble(read.csv(paste0(cwd, "/../data/input/daily_revenue-challenge.csv")))
```

Data Cleansing

The functions below, as well as the dplyr verbs, were carefully chosen through experimentation to get the data to present itself in useful formats respecting the tidy data standard.

listings-challenge.csv noteworthy transformations

Category and number of rooms are separated so it is possible to analyse these predictors individually. Address column is dropped since it is unstructured data which signal is assumed to be captured by Localization variable. The strange "TOPM" category merges with "TOP" category.

daily_revenue-challenge.csv noteworthy transformations

Reservation advance variable introduced as it is regarded as a key metric

```
# Auxiliary cleaning functions
parse_double_with_comma <- function(x){
  parse_number(x, locale=locale(decimal_mark=","))
}
parse_integer_with_comma <- function(x){
  as.integer(parse_double_with_comma(x))
}

# Prepares datasets to perform relevant analysis
tidy.listings <- listings %>%
  mutate(across(c("Tipo", "Status", "Hotel", "Categoria", "Localização"),
                as.factor)) %>%
  mutate(across(c("Comissão", "Banheiros", "Taxa.de.Limpeza"),
                parse_double_with_comma)) %>%
  mutate(across(c(contains("Cama"), "Travesseiros", "Capacidade"),
                parse_integer_with_comma)) %>%
  mutate(Data.Inicial.do.contrato=dmy(Data.Inicial.do.contrato)) %>%
  extract(Categoria, c("Categoria", "Quartos"), "[HOU]*([A-Z]+)([0-9])*Q*",
          convert=TRUE) %>%
  mutate(Categoria=as.factor(Categoria)) %>%
```

```
  select(-c("Endereço")) %>%
  mutate(Categoria=fct_collapse(Categoria, TOP=c("TOP", "TOPM")))

tidy.daily.revenue <- daily.revenue %>%
  mutate(listing=as.factor(listing)) %>%
  mutate(across(contains("date"), ~as_date(parse_datetime(.)))) %>%
  mutate(reservation_advance=date - creation_date)

tidy.listings
```

```
## # A tibble: 462 x 18
##    Código Localização Categoria Quartos Comissão Cama.Casal Cama.Solteiro
##    <chr>  <fct>       <fct>       <int>    <dbl>      <int>         <int>
##  1 ADJ205 JUR         SIM             2      0.2          1             2
##  2 AYA201 JUR         JR              3      0.2          3             0
##  3 BEL102 JUR         SUP             2      0.2          0             2
##  4 CCA201 CAN         SUP             3      0.2          1             4
##  5 CDC301 JUR         TOP             3      0.2          3             0
##  6 CDS106 JUR         SUP             2      0.2          1             2
##  7 CDS208 JUR         JR              2      0.2          1             2
##  8 CDS209 JUR         SUP             2      0.2          1             2
##  9 CHR101 JUR         SUP             2      0.2          1             2
## 10 CPJ102 JUR         TOP             3      0.2          2             2
## # ... with 452 more rows, and 11 more variables: Cama.Queen <int>,
## #   Cama.King <int>, Sofá.Cama.Solteiro <int>, Travesseiros <int>,
## #   Banheiros <dbl>, Taxa.de.Limpeza <dbl>, Capacidade <int>, Hotel <fct>,
## #   Data.Inicial.do.contrato <date>, Status <fct>, Tipo <fct>
```

```
tidy.daily.revenue
```

```
## # A tibble: 289,923 x 8
##    listing date       last_offered_price occupancy revenue blocked creation_date
##    <fct>   <date>                  <dbl>     <dbl>   <dbl>   <int> <date>
##  1 ABC102  2021-12-24                  0         1       0       1 2021-12-24
##  2 ABC102  2021-12-25                  0         0       0       0 NA
##  3 ABC102  2021-12-26                  0         0       0       0 NA
##  4 ABC102  2021-12-27                  0         0       0       0 NA
##  5 ABC102  2021-12-28               2248         1    2248       0 2021-12-24
##  6 ABC102  2021-12-29               2248         1    2248       0 2021-12-24
##  7 ABC102  2021-12-30               2248         1    2248       0 2021-12-24
##  8 ABC102  2021-12-31               2248         1    2248       0 2021-12-24
##  9 ABC102  2022-01-01               2248         1    2248       0 2021-12-24
## 10 ABC102  2022-01-02               2248         1    2248       0 2021-12-24
## # ... with 289,913 more rows, and 1 more variable: reservation_advance <drtn>
```

Data Agreggation

In order to train predictive models, joining the two datasets is desired.

```
daily.revenue.listings <- tidy.daily.revenue %>%
  left_join(tidy.listings, by=c("listing" = "Código")) %>%
  mutate(commission=last_offered_price*Comissão,
         listing=as.factor(listing))

daily.revenue.listings
```

```
## # A tibble: 289,923 x 26
##    listing date       last_offered_price occupancy revenue blocked creation_date
##    <fct>   <date>                  <dbl>     <dbl>   <dbl>   <int> <date>
##  1 ABC102  2021-12-24                  0         1       0       1 2021-12-24
##  2 ABC102  2021-12-25                  0         0       0       0 NA
##  3 ABC102  2021-12-26                  0         0       0       0 NA
##  4 ABC102  2021-12-27                  0         0       0       0 NA
##  5 ABC102  2021-12-28               2248         1    2248       0 2021-12-24
##  6 ABC102  2021-12-29               2248         1    2248       0 2021-12-24
##  7 ABC102  2021-12-30               2248         1    2248       0 2021-12-24
##  8 ABC102  2021-12-31               2248         1    2248       0 2021-12-24
##  9 ABC102  2022-01-01               2248         1    2248       0 2021-12-24
## 10 ABC102  2022-01-02               2248         1    2248       0 2021-12-24
## # ... with 289,913 more rows, and 19 more variables:
## #   reservation_advance <drtn>, Localização <fct>, Categoria <fct>,
## #   Quartos <int>, Comissão <dbl>, Cama.Casal <int>, Cama.Solteiro <int>,
## #   Cama.Queen <int>, Cama.King <int>, Sofá.Cama.Solteiro <int>,
## #   Travesseiros <int>, Banheiros <dbl>, Taxa.de.Limpeza <dbl>,
## #   Capacidade <int>, Hotel <fct>, Data.Inicial.do.contrato <date>,
## #   Status <fct>, Tipo <fct>, commission <dbl>
```

Exploratory Data Analysis

Check for NAs

```
sapply(daily.revenue.listings, function(x) sum(is.na(x)))
```

```
##                  listing                     date       last_offered_price
##                        0                        0                        0
##                occupancy                  revenue                  blocked
##                        0                        0                        0
##            creation_date      reservation_advance              Localização
##                   208288                   208288                      902
##                Categoria                  Quartos                 Comissão
##                      902                   113630                      902
##               Cama.Casal            Cama.Solteiro               Cama.Queen
##                      902                      902                      902
##                Cama.King       Sofá.Cama.Solteiro             Travesseiros
##                      902                      902                     2207
##                Banheiros          Taxa.de.Limpeza               Capacidade
##                      902                      902                      902
##                    Hotel Data.Inicial.do.contrato                   Status
##                      902                      902                      902
##                     Tipo               commission
##                      902                      902
```

The NA count suggests there are listings in daily_revenue.csv not in listings.csv

```
unique.listings <- as.character(unique(tidy.daily.revenue$listing))
unique.listings[-which(unique.listings %in% tidy.listings$Código)]
```

```
## [1] "TST001"
```

Indeed, checking the TST001 rows in daily.revenue.listings yields:

```r
daily.revenue.listings[which(daily.revenue.listings$listing == "TST001"),] %>%
  select(listing, setdiff(names(tidy.listings), c("Código")))
```

```
## # A tibble: 902 x 18
##    listing Localização Categoria Quartos Comissão Cama.Casal Cama.Solteiro
##    <fct>   <fct>       <fct>       <int>    <dbl>      <int>         <int>
##  1 TST001  <NA>        <NA>           NA       NA         NA            NA
##  2 TST001  <NA>        <NA>           NA       NA         NA            NA
##  3 TST001  <NA>        <NA>           NA       NA         NA            NA
##  4 TST001  <NA>        <NA>           NA       NA         NA            NA
##  5 TST001  <NA>        <NA>           NA       NA         NA            NA
##  6 TST001  <NA>        <NA>           NA       NA         NA            NA
##  7 TST001  <NA>        <NA>           NA       NA         NA            NA
##  8 TST001  <NA>        <NA>           NA       NA         NA            NA
##  9 TST001  <NA>        <NA>           NA       NA         NA            NA
## 10 TST001  <NA>        <NA>           NA       NA         NA            NA
## # ... with 892 more rows, and 11 more variables: Cama.Queen <int>,
## #   Cama.King <int>, Sofá.Cama.Solteiro <int>, Travesseiros <int>,
## #   Banheiros <dbl>, Taxa.de.Limpeza <dbl>, Capacidade <int>, Hotel <fct>,
## #   Data.Inicial.do.contrato <date>, Status <fct>, Tipo <fct>
```

Since 0 revenue is made from this listing, it is safe to drop its rows without compromising future results.

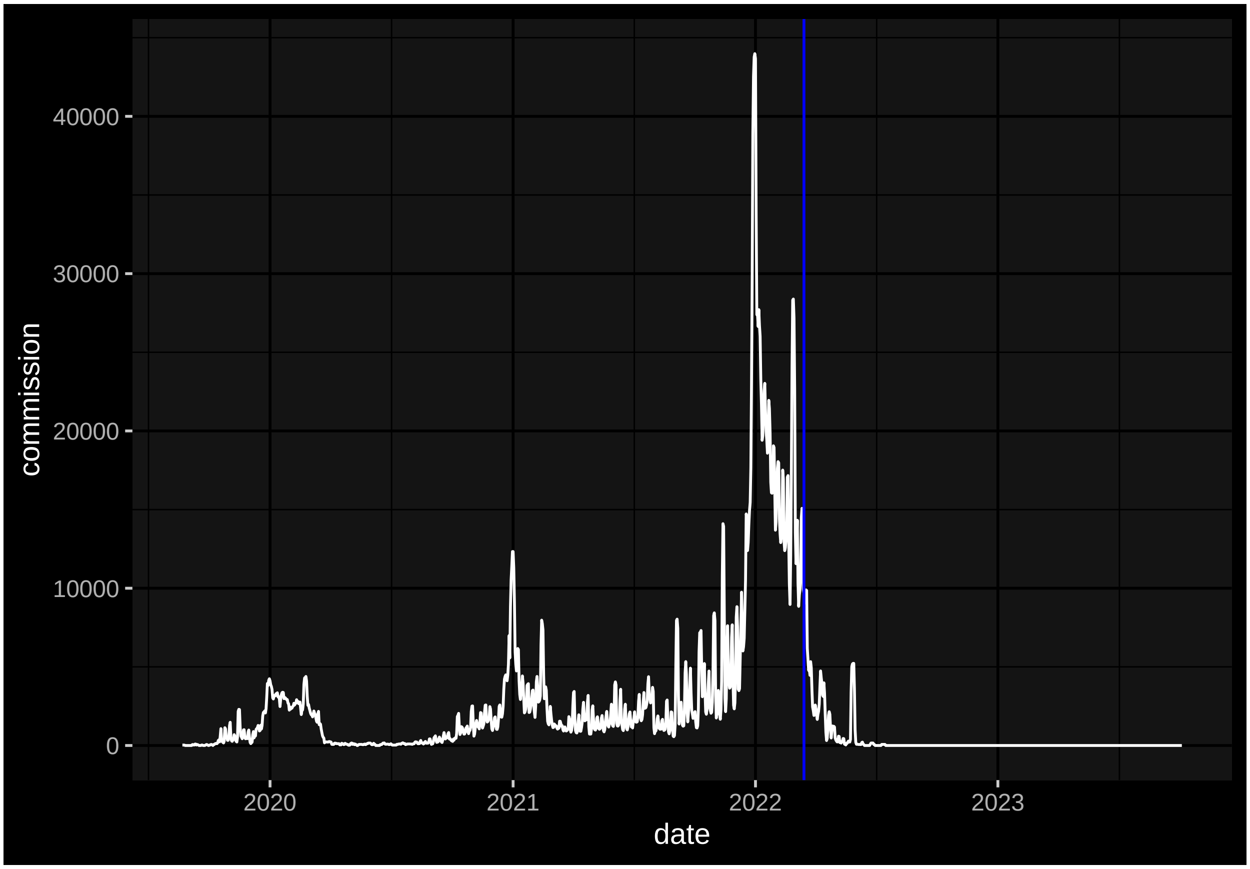
```r
sum(daily.revenue.listings[which(daily.revenue.listings$listing == "TST001"),]$revenue)
```

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

```r
daily.revenue.listings <- filter(daily.revenue.listings, listing != "TST001")
```

Commission earned across time

```r
daily.revenue.listings %>%
  mutate(commission=revenue*Comissão) %>%
  select(date, commission) %>%
  group_by(date) %>%
  summarise(commission=sum(commission)) %>%
  ggplot(aes(x=date, y=commission)) +
  geom_line() +
  geom_vline(xintercept=as.numeric(ymd("2022-03-15")), colour="blue")
```



The COVID-19 pandemic effects can be seen clearly in the plot. To investigate it a little further:

```
daily.revenue.listings %>%
  filter(date >= "2020-01-01", date <= "2020-12-31")
```

```
## # A tibble: 23,388 x 26
##    listing date       last_offered_price occupancy revenue blocked creation_date
##    <fct>   <date>                  <dbl>     <dbl>   <dbl>   <int> <date>
##  1 ADJ205  2020-01-01                  0         1       0       1 2019-11-21
##  2 ADJ205  2020-01-02                  0         1       0       1 2019-11-21
##  3 ADJ205  2020-01-03                  0         1       0       1 2019-11-21
##  4 ADJ205  2020-01-04                  0         1       0       1 2019-11-21
##  5 ADJ205  2020-01-05                  0         1       0       1 2019-11-21
##  6 ADJ205  2020-01-06                  0         1       0       1 2019-11-21
##  7 ADJ205  2020-01-07                  0         1       0       1 2019-11-21
##  8 ADJ205  2020-01-08                  0         1       0       1 2019-11-21
##  9 ADJ205  2020-01-09                  0         1       0       1 2019-11-21
## 10 ADJ205  2020-01-10                  0         1       0       1 2019-11-21
## # ... with 23,378 more rows, and 19 more variables: reservation_advance <drtn>,
## #   Localização <fct>, Categoria <fct>, Quartos <int>, Comissão <dbl>,
## #   Cama.Casal <int>, Cama.Solteiro <int>, Cama.Queen <int>, Cama.King <int>,
## #   Sofá.Cama.Solteiro <int>, Travesseiros <int>, Banheiros <dbl>,
## #   Taxa.de.Limpeza <dbl>, Capacidade <int>, Hotel <fct>,
## #   Data.Inicial.do.contrato <date>, Status <fct>, Tipo <fct>, commission <dbl>
```

It is relevant to notice that all entries were generated up to a date:

```
max(daily.revenue.listings$creation_date, na.rm=TRUE)
```

```
## [1] "2022-03-15"
```

As a consequence, all revenue posterior to it (blue vertical line) is not yet consolidated.

Understanding key variables influencing revenue

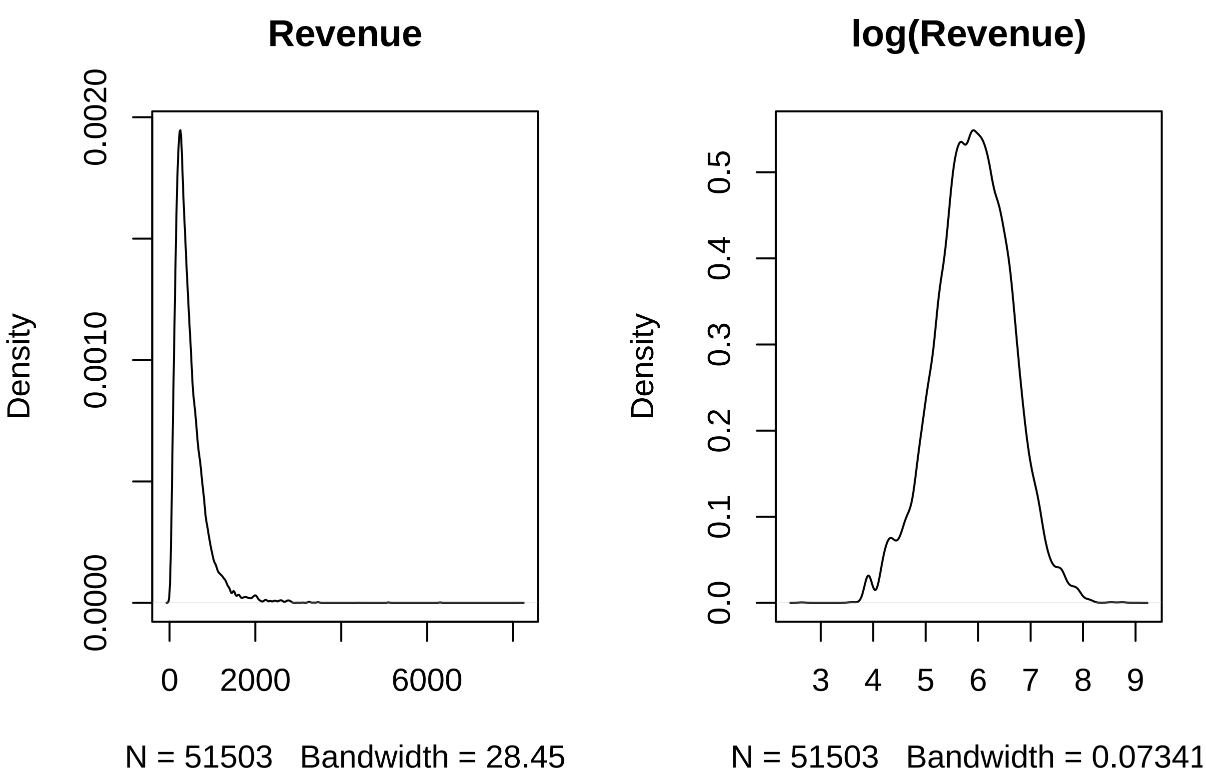
A log transform is applied in order to best account for variation scale in revenue variable.

```
revenue.made.data <- daily.revenue.listings %>%
  # Feature engineering applied here
  mutate(is.weekend = timeDate::isWeekend(date),
         is.holiday = date %in% holidaysANBIMA) %>%
  filter(occupancy==1, blocked==0) %>%
  # Adjusts scale to enhance model fitting
  mutate(log.revenue = log(revenue)) %>%
  # Ignores redundant or not relevant predictors
  select(-c(last_offered_price, listing, date, occupancy, blocked, Comissão,
            creation_date, Data.Inicial.do.contrato, Travesseiros, Status,
            commission, revenue))
```

Comparing revenue against log(revenue) distributions

```
par(mfrow=c(1, 2))

plot(density(exp(revenue.made.data$log.revenue)), main="Revenue")
plot(density(revenue.made.data$log.revenue), main="log(Revenue)")
```



Checking for NAs

```
sapply(revenue.made.data, function(x) sum(is.na(x)))
```

```
## reservation_advance          Localização           Categoria             Quartos
##                   0                    0                   0               23357
##          Cama.Casal        Cama.Solteiro          Cama.Queen           Cama.King
##                   0                    0                   0                   0
##  Sofá.Cama.Solteiro            Banheiros     Taxa.de.Limpeza          Capacidade
##                   0                    0                   0                   0
##               Hotel                 Tipo          is.weekend          is.holiday
##                   0                    0                   0                   0
##         log.revenue
##                   0
```

```
revenue.made.data.missing.rooms <- revenue.made.data[
  which(is.na(revenue.made.data$Quartos)),
]
all(revenue.made.data.missing.rooms$Hotel == "Sim")
```

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

Since some variables behave always differently given the listing is or is not inside a hotel, two models are made to investigate predictor importance on both occasions.

Hotel and No Hotel split

```
revenue.made.data.hotel <- revenue.made.data %>%
  filter(Hotel == "Sim") %>%
  select(-c(Hotel, Quartos, Taxa.de.Limpeza, Tipo))

revenue.made.data.no.hotel <- revenue.made.data %>%
  filter(Hotel == "Não") %>%
  mutate(log.Taxa.de.Limpeza = log(Taxa.de.Limpeza)) %>%
  select(-c(Hotel, Taxa.de.Limpeza))

head(revenue.made.data.hotel)
```

```
## # A tibble: 6 x 13
##   reservation_advance Localização Categoria Cama.Casal Cama.Solteiro Cama.Queen
##   <drtn>              <fct>       <fct>          <int>         <int>      <int>
## 1  3 days             ILC         TOP                0             0          0
## 2  4 days             ILC         TOP                0             0          0
## 3  5 days             ILC         TOP                0             0          0
## 4 12 days             ILC         TOP                0             0          0
## 5 13 days             ILC         TOP                0             0          0
## 6 14 days             ILC         TOP                0             0          0
## # ... with 7 more variables: Cama.King <int>, Sofá.Cama.Solteiro <int>,
## #   Banheiros <dbl>, Capacidade <int>, is.weekend <lgl>, is.holiday <lgl>,
## #   log.revenue <dbl>
```

```
head(revenue.made.data.no.hotel)
```

```
## # A tibble: 6 x 16
##   reservation_advance Localização Categoria Quartos Cama.Casal Cama.Solteiro
##   <drtn>              <fct>       <fct>       <int>      <int>         <int>
```

```
## 1 4 days             ITA         MASTER          3          2             2
## 2 5 days             ITA         MASTER          3          2             2
## 3 6 days             ITA         MASTER          3          2             2
## 4 7 days             ITA         MASTER          3          2             2
## 5 8 days             ITA         MASTER          3          2             2
## 6 9 days             ITA         MASTER          3          2             2
## # ... with 10 more variables: Cama.Queen <int>, Cama.King <int>,
## #   Sofá.Cama.Solteiro <int>, Banheiros <dbl>, Capacidade <int>, Tipo <fct>,
## #   is.weekend <lgl>, is.holiday <lgl>, log.revenue <dbl>,
## #   log.Taxa.de.Limpeza <dbl>
```

Best subset selection for linear regression

```r
summary(step(lm(log.revenue ~ ., data=revenue.made.data.hotel), trace=0))
```

```
##
## Call:
## lm(formula = log.revenue ~ reservation_advance + Localização +
##     Categoria + Cama.Casal + Cama.Queen + Sofá.Cama.Solteiro +
##     Banheiros + Capacidade + is.weekend + is.holiday, data = revenue.made.data.hotel)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.43455 -0.39619 -0.00043  0.35240  3.10246
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.812e+00  3.358e-02 173.094  < 2e-16 ***
## reservation_advance  1.100e-03  9.036e-05  12.177  < 2e-16 ***
## LocalizaçãoJBV      -3.084e-01  8.437e-03 -36.557  < 2e-16 ***
## LocalizaçãoSLA      -1.243e+00  1.378e-01  -9.022  < 2e-16 ***
## LocalizaçãoSTO       3.569e-01  3.037e-02  11.754  < 2e-16 ***
## CategoriaMASTER      4.393e-01  1.858e-02  23.644  < 2e-16 ***
## CategoriaSUP         1.080e-01  2.853e-02   3.784 0.000155 ***
## CategoriaTOP         1.589e-01  8.719e-03  18.223  < 2e-16 ***
## Cama.Casal           1.051e-01  4.814e-02   2.184 0.029005 *
## Cama.Queen           3.340e-01  5.025e-02   6.646 3.07e-11 ***
## Sofá.Cama.Solteiro   8.029e-02  9.971e-03   8.052 8.50e-16 ***
## Banheiros           -2.864e-01  5.079e-02  -5.638 1.74e-08 ***
## Capacidade           9.389e-02  8.481e-03  11.070  < 2e-16 ***
## is.weekendTRUE      -2.030e-02  7.302e-03  -2.780 0.005441 **
## is.holidayTRUE       2.551e-01  1.654e-02  15.426  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5331 on 23662 degrees of freedom
## Multiple R-squared:  0.1352, Adjusted R-squared:  0.1347
## F-statistic: 264.2 on 14 and 23662 DF,  p-value: < 2.2e-16
```

```r
summary(step(lm(log.revenue ~ ., data=revenue.made.data.no.hotel), trace=0))
```

```
##
## Call:
## lm(formula = log.revenue ~ Localização + Categoria + Quartos +
##     Cama.Casal + Cama.Solteiro + Cama.Queen + Cama.King + Sofá.Cama.Solteiro +
```

```
##     Banheiros + Tipo + is.weekend + is.holiday + log.Taxa.de.Limpeza,
##     data = revenue.made.data.no.hotel)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.89354 -0.34849  0.03672  0.37830  1.71847
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.133927   0.238850  25.681  < 2e-16 ***
## LocalizaçãoBOM       0.317843   0.039028   8.144 3.98e-16 ***
## LocalizaçãoCAM      -0.022619   0.041666  -0.543   0.5872
## LocalizaçãoCAN      -0.264805   0.035729  -7.411 1.28e-13 ***
## LocalizaçãoCEN      -0.329763   0.038663  -8.529  < 2e-16 ***
## LocalizaçãoCON      -0.362220   0.051255  -7.067 1.62e-12 ***
## LocalizaçãoGRA      -0.152737   0.081306  -1.879   0.0603 .
## LocalizaçãoING      -0.305292   0.034535  -8.840  < 2e-16 ***
## LocalizaçãoITA       0.434543   0.051816   8.386  < 2e-16 ***
## LocalizaçãoITP      -0.162489   0.092911  -1.749   0.0803 .
## LocalizaçãoJUR      -0.076360   0.033764  -2.262   0.0237 *
## LocalizaçãoLAG      -0.447980   0.036639 -12.227  < 2e-16 ***
## LocalizaçãoPBL       0.375435   0.066345   5.659 1.54e-08 ***
## LocalizaçãoSAN      -0.478635   0.088295  -5.421 5.98e-08 ***
## LocalizaçãoTBM       0.281795   0.124848   2.257   0.0240 *
## LocalizaçãoUFSC     -0.354180   0.036696  -9.652  < 2e-16 ***
## CategoriaMASTER      0.684433   0.018595  36.808  < 2e-16 ***
## CategoriaSIM         0.020292   0.014430   1.406   0.1597
## CategoriaSUP         0.237794   0.010488  22.672  < 2e-16 ***
## CategoriaTOP         0.356288   0.010764  33.099  < 2e-16 ***
## Quartos              0.346304   0.015261  22.692  < 2e-16 ***
## Cama.Casal           0.025833   0.012643   2.043   0.0410 *
## Cama.Solteiro       -0.034575   0.007466  -4.631 3.66e-06 ***
## Cama.Queen           0.093178   0.013534   6.885 5.91e-12 ***
## Cama.King            0.271441   0.028156   9.641  < 2e-16 ***
## Sofá.Cama.Solteiro   0.039911   0.006642   6.009 1.89e-09 ***
## Banheiros            0.048872   0.008291   5.894 3.80e-09 ***
## TipoCasa             0.110881   0.019810   5.597 2.20e-08 ***
## is.weekendTRUE       0.034163   0.007365   4.638 3.53e-06 ***
## is.holidayTRUE       0.289770   0.016656  17.397  < 2e-16 ***
## log.Taxa.de.Limpeza -0.231567   0.047402  -4.885 1.04e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5703 on 27795 degrees of freedom
## Multiple R-squared:  0.3535, Adjusted R-squared:  0.3529
## F-statistic: 506.7 on 30 and 27795 DF,  p-value: < 2.2e-16
```

The low values for R^2 and large residuals standard errors suggests that a linear regression model is not well suited to the data at hand.

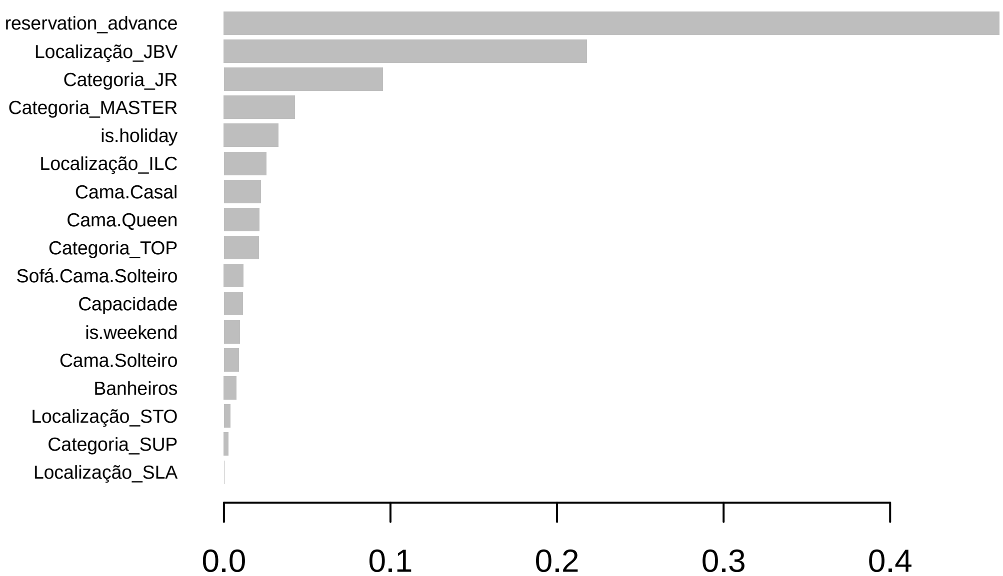
Boosted Trees Regression (Xtreme Gradient Boosting)

Adopting a more robust model, proven to work in similar cases, yields:

```
hotel.matrix <- as.matrix(revenue.made.data.hotel %>%
                            select(-c(log.revenue)) %>%
                            mutate(reservation_advance=as.numeric(reservation_advance)) %>%
                            dummy_cols(remove_selected_columns = TRUE))
model.hotel <- xgboost(data=hotel.matrix,
                       label=revenue.made.data.hotel$log.revenue,
                       nrounds=20,
                       max.depth=6)
```

```
## [1]  train-rmse:4.049695
## [2]  train-rmse:2.859090
## [3]  train-rmse:2.034334
## [4]  train-rmse:1.469064
## [5]  train-rmse:1.088740
## [6]  train-rmse:0.840704
## [7]  train-rmse:0.686698
## [8]  train-rmse:0.595840
## [9]  train-rmse:0.545332
## [10] train-rmse:0.518455
## [11] train-rmse:0.504216
## [12] train-rmse:0.496376
## [13] train-rmse:0.492548
## [14] train-rmse:0.490101
## [15] train-rmse:0.489029
## [16] train-rmse:0.488322
## [17] train-rmse:0.487676
## [18] train-rmse:0.486954
## [19] train-rmse:0.486467
## [20] train-rmse:0.485744
```

```
xgb.plot.importance(xgb.importance(model=model.hotel))
```

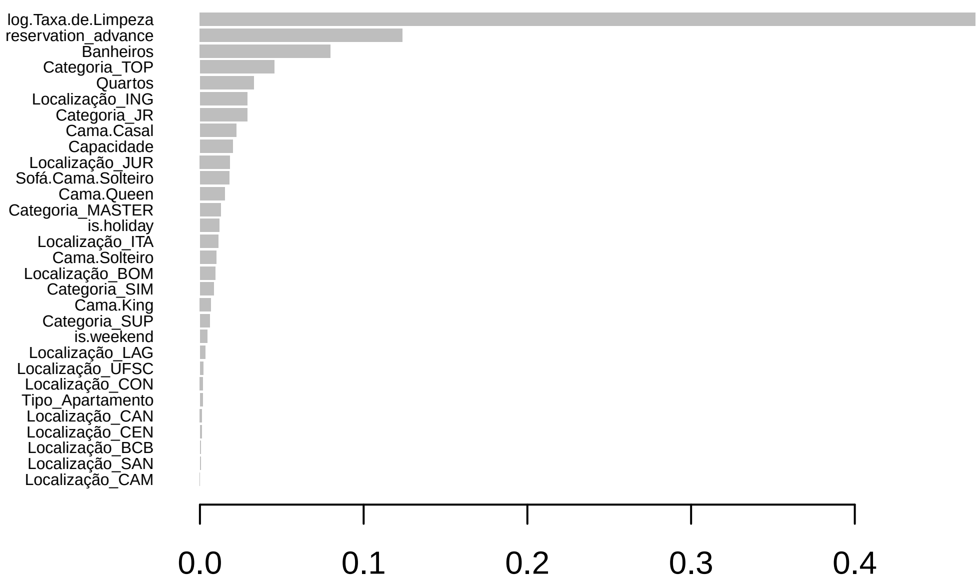



```
no.hotel.matrix <- as.matrix(revenue.made.data.no.hotel %>%
                               select(-c(log.revenue)) %>%
                            mutate(reservation_advance=as.numeric(reservation_advance)) %>%
                            dummy_cols(remove_selected_columns = TRUE))
```

```r
model.no.hotel <- xgboost(data=no.hotel.matrix,
                         label=revenue.made.data.no.hotel$log.revenue,
                         nrounds=20,
                         max.depth=6)
```

```
## [1]  train-rmse:3.648831
## [2]  train-rmse:2.583226
## [3]  train-rmse:1.846882
## [4]  train-rmse:1.345623
## [5]  train-rmse:1.007995
## [6]  train-rmse:0.791748
## [7]  train-rmse:0.657365
## [8]  train-rmse:0.578725
## [9]  train-rmse:0.535778
## [10] train-rmse:0.511035
## [11] train-rmse:0.497317
## [12] train-rmse:0.489923
## [13] train-rmse:0.485691
## [14] train-rmse:0.483402
## [15] train-rmse:0.481763
## [16] train-rmse:0.478763
## [17] train-rmse:0.477714
## [18] train-rmse:0.474522
## [19] train-rmse:0.472613
## [20] train-rmse:0.471760
```

```r
xgb.plot.importance(xgb.importance(model=model.no.hotel))
```



Export data to next step

```r
write_csv(daily.revenue.listings,
          paste0(cwd, "/../data/output/daily_revenue_listings.csv"))
```