# Price prediction for Airbnb Listings in Bangkok

Bruno Helmeczy

02/02/2021

## Abstract

This report investigates how to predict the prices at which apartments in Bangkok can be rented out on Airbnb. Below I discuss a number of features taken into account, compare 6 prediction models of varying complexity, & after choosing & re-estimating my final model, I investigate expected performance across property types, the number of people accommodated, while investigating most important accommodation features. Based on 5-fold cross validation, I found the expanded Random Forest model predicting log-transformed prices to perform best in terms of all metrics. After re-estimation, the final model boasted 56.1% R-squared, & performed with 17 dollar MAE & 33.2 dollar RMSE, ca. 67% of the average price during out-of-sample testing. Finally, model diagnostics showed the model to perform fairly consistently given accommodation size (2-6 people) & most frequent neighbourhoods, yet RMSE decreased to 56% of average price when predicting apartments, vs 73% for condominium property types.

### Data, Cleaning & Feature Engineering

The raw dataset was downloaded from- & is available at AirBnB-s website (from here), & comprises over 19.7K observations & 74 Variables, summarizing all accommodations in the Bangkok area available for rental, as of 23rd-24th December, 2020. After keeping only apartments-, & condominiums hosting between 2-6 people, 9962 observations remained to clean for analysis. To manage this feature space, i.e. to keep what's important & drop what is not, while looking to maintain intuition throughout the process, I grouped variables into 7 subjects (excluding ID variables): **Host** information, **Geo-Spatial**-, & **Property** information, **Sales**-, & **Availability**-related data, & **Reviews**-, & **Listings** data.

Some of these groups proved rather not useful, e.g. from **Geo-spatial** variable group only the **neighbourhood_cleansed** variable is retained, as latitudinal-, & longitudinal metrics are indirectly already captured by the neighbourhood they fall into, while the raw **neighbourhood** variable held 500+ values, also posing a linguistic challenge. Similarly, **Listings**-group variables were in essence all dropped, as they contain information regarding listings of different room/apartment types, or contain information already present in **Host**-related variables. My substantive rationale for that is **a)** the number of listings attributed to a host above all are a proxy for the likelihood of him/her being a scammer, rather then an attribute impacting rentals' pricing, & **b)** even on AirBnb, this information is presented only as a host having either 1, or multiple listings online. As such, seeing as prices are rather market-, then inventory-based, a binary variable showing whether the host has only 1 listing seems sufficient. To be safe, the integer variable denoting how many listings a host has, is also kept, so a LASSO model could capture if either is less important by dropping it.

Speaking of **Host**-variables, I view their primary purpose is indicating whether a listing is a scam or not, as these are what both Airbnb & prospective guests use this information for. Thus, 6 of 8 Host-variables are logical / binary variables, while host verifications are a list of credential details (similarly to amenities being a list of objects in the rental property). Any single 1 of these credential details is not important, say whether a host has an email address or Facebook profile, however the total number of credentials may indicate hosts' credibility, & professionalism, so I simply counted the **number of verifications** a host provided.

**Guest Reviews**-related variables: As proxy for a hosts' experience & recent activity, Nr. of days since their 1st of last reviews were calculated. Ca. 34% NAs, for hosts without reviews - seen from the same observations also missing review scores, & Nr. of reviews per month. In terms of review ratings, only the overall review scores are kept, being an aggregation of sub-scores, & for Nr. of days since 1st & last reviews, instead of the median-, the maximum values were imputed, signifying the quesi-infinite number of days the host has had a review. To impute NAs for review scores, I took the arithmetic mean instead of the median. Though the mean is less robust to extreme values, there is some level of uncertainty arising in guests' minds, if seeing there are no reviews what so ever for a listing. I intend to take this into account by using the mean.

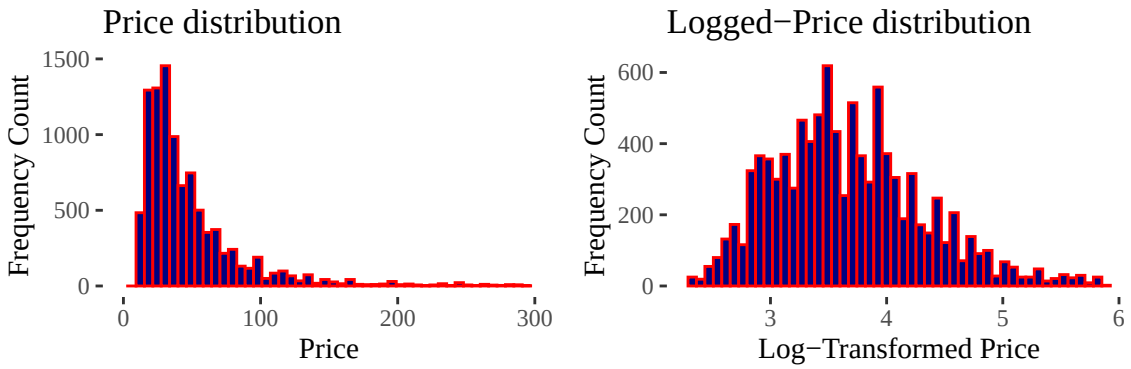
**Sales & Availability**-related variables are what hoteliers would consider most obviously relating to prices, & besides price, also include minimum-maximum stay-restrictions, as well as the number of available room nights in the coming periods of different length. I argue the number of room nights a host has sold more closely relates to what price he asks for, rather then the number of room nights he has available in the future, given the confidence it may instill. So, I transformed these to denote the number of rooms sold in future periods of different lengths, & eventually reduced these to factor variables. With stay-restrictions, I 1st wrote a function to calculate how identical these columns are, allowing me eventually to retain only minimum stay nights, as maximum nights contained many extreme-, & missing values.

**Property Variables** The final, & perhaps most important feature engineering decision is regarding **Property**-related variables, most strongly contributing the dropping observations, & 48 dummy variables standing for amenities. To obtain data only for apartments, I 1st filtered the room_type variable to only include entire homes or apartments, & also removed listings where the property type was described as e.g. Room, Castle, hostel, or treehouse & other properties of similarly different nature. This still left some non-trivial property types, like Bungalows, Guest Suites, Lofts & Villas, but I view these as legitimate alternatives in potential guests' minds, whereas I hazard, the type of guest looking to book a Castle or a Treehouse is materially different then someone looking to stay in apartments. Once filtered for these property types, listings priced above 350 USD were also dropped, resulting in the cleaned dataset holding 9866 observations.

The greatest cleaning challenge however, is of handling the **amenities** column. After creating a dummy-variable table of all possible amenities, I wrote 3 functions to handle this variable: 1st to coerce columns whose names match certain sets, & combinations of keywords; 2nd to remove duplicate columns, & 3rd to check the frequency of Yes-es in the remaining dummy columns, latter looking to remove amenities with imbalanced values, by myself defined with having less then 1% Yes values. All this resulted in reducing the number of amenities dummies from 262 to 48.

## Modelling

Eventually, I grouped my variables into **Property-**, **Host-**, **Reviews-**, & **Amenities**-related, seeing these as the varying degrees, & order of additional information a guest might be looking for, to decide whether to stay at a given listed rental or not. For OLS & LASSO models, I also incorporated interactions between key property-related variables & all amenities, & predicted log-transformed price in 5 of 6 models, seeing how skewed it is. Please see the finalized variable groups, & derived predictor groupings in bulleted lists below:



- **Property**-related: Property Type, Nr. of Guests accommodated (level, log-,level squared, log-squared), Nr. of Beds-, Bedrooms & Bathrooms, Neighbourhood Cleansed, Has Availability, Nr. of Rooms Sold in coming 30-, 60-, 90-, & 365 days.
- **Host**-related: Host is owner, Nr. of hosts' listings & verifications, & whether he greets you, accept every reservation, has a profile pic, & his/her identity is verified.
- **Reviews**-related: Nr. of Reviews (Total & per month), Average Review score, Nr. of days since 1st & last reviews (Level- & Log form), & Flag-variables for missing observations on review per month, score rating, & Nr. of days since last review variables.
- **Amenities:**-related: 48 dummy variables indicating whether a collection of amenities are present.
- **Interactions:** I hazard the presence of certain amenities affects rental prices to varying degrees depending on property characteristics, e.g. though wifi is a given necessity in any context, it's perhaps even more important in larger accommodations to keep children at peace. Thus, all 48 amenities were interacted with core property features, Nr. of beds / bedrooms / bathroooms & people accommodated.
- **Predictor Group 1:** Property Variable group
- **Predictor Group 2:** Predictor Group 1 + Host-, & Reviews variable groups
- **Predictor Group 3:** Predictor Group 2 + Amenities
- **Predictor Group E:** Predictor Group 3 + Interactions

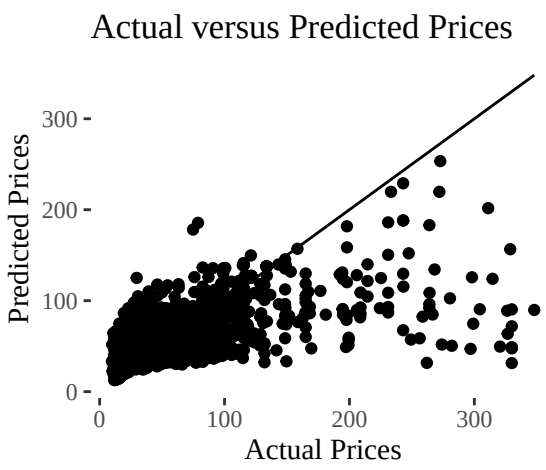
Finally, please see the created Predictor groups above. In the 4 best models summarized below, LASSO utilizes Predictor Group E, the simplified Random Forest (rf_model_1) used Predictor Group 2, & the rest Predictor Group 3. Due to prices' strong skewness shown earlier, all models predict log-prices, which were subsequently transformed, to obtain price predictions & calculate MAE & RMSE.

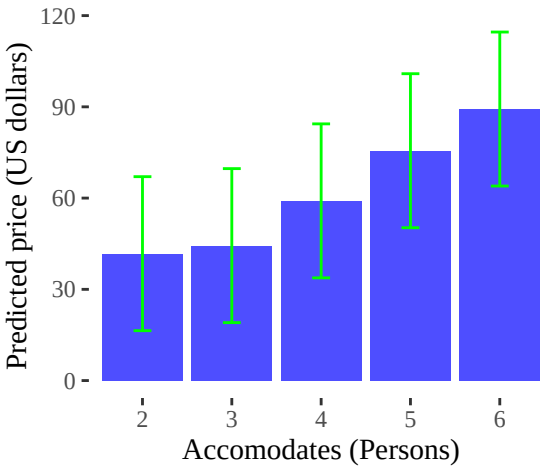
| models | Rsq | MAE | RMSE | RMSE_norm |
|---|---|---|---|---|
| ols_model | 0.437 | 19.035 | 33.7 | 0.678 |
| lasso_model | 0.478 | 17.958 | 32.227 | 0.648 |
| rf_model_1 | 0.528 | 10.169 | 22.346 | 0.449 |
| rf_model_2 | 0.56 | 8.875 | 20.194 | 0.406 |
| rf_model_2_level | 0.46 | 18.416 | 31.902 | 0.642 |

As can be seen above, the extended Random Forest model (rf_model_2) beat all other candidates, with 5-fold cross-validated performance of R-sqaured of 56%, & a RMSE of 20.2 USD, ca. 40% of the average listing price. As such, I recommend the extended Random Forest to estimate the clients' future listings' profitability. What's left is re-estimating the chosen model on the complete training dataset & test expected future performance ex-sample, utilizing data reserved only for this stage, 30% of the cleaned dataset. As visible below, the model performs substantially worse ex-sample, on par with LASSO's cross-validated performance, though beating both regression models in terms of Mean Average Error. This indicates a number of especially bad predictions to cause worse metrics for the final model, something expected when tuning models to predict log-prices & transforming them to raw price predictions.

| modelname | Rsq | MAE | RMSE | RMSE_norm |
|---|---|---|---|---|
| Final Full Random Forest Model | 0.561 | 17.3 | 33.2 | 0.67 |
| Full Random Forest w Level Price | 0.472 | 17.8 | 32.6 | 0.656 |

## Model Implications

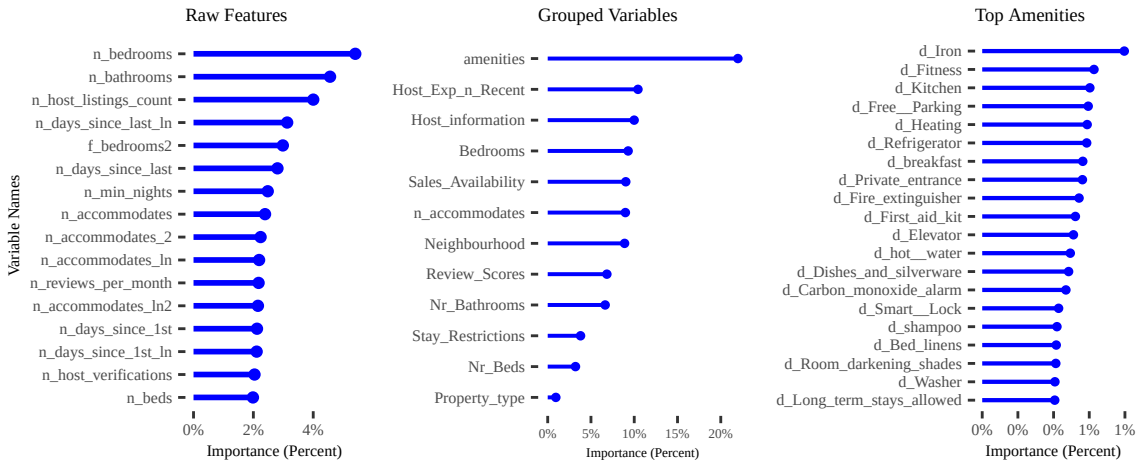


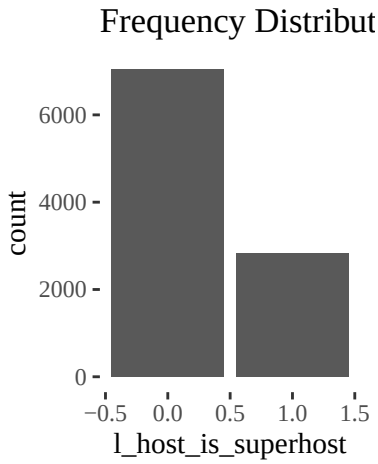

Please see the plots of predicted prices scattered around the line indicating actual prices. Visibly, the chosen model tends to predict right on average up to 100 USD, however is very strongly under predicting prices above 100 USD, not having a single case of over-prediction above ca. 133 USD. One can see however, that the same model predicting level prices

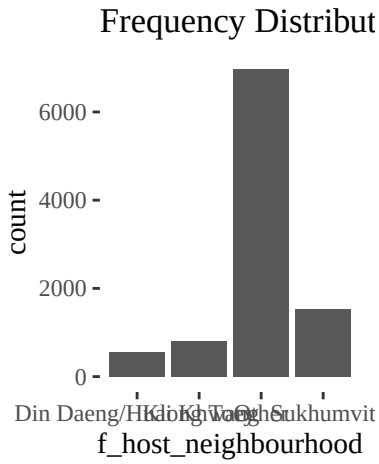
I nevertheless maintain the models' validity, having cross-validated each of 6 models, including the chosen extend Random Forest model predicting level prices, which performed roughly twice as bad on the holdout set (also shown below, for reference). On the other hand, from a business perspective this means the chosen model can confidently estimate price ranges lower bounds, more useful to evaluate investment decisions.

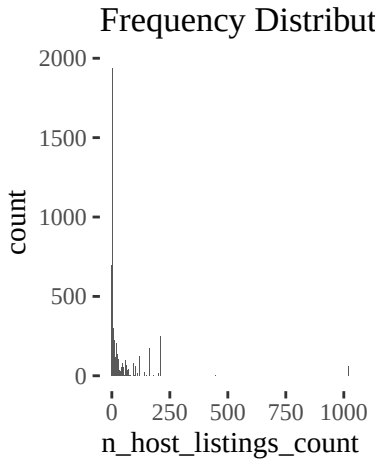
### Variable Importance

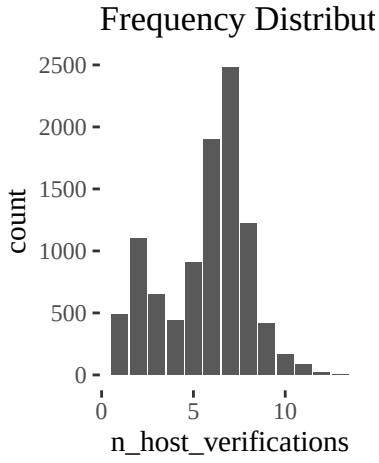


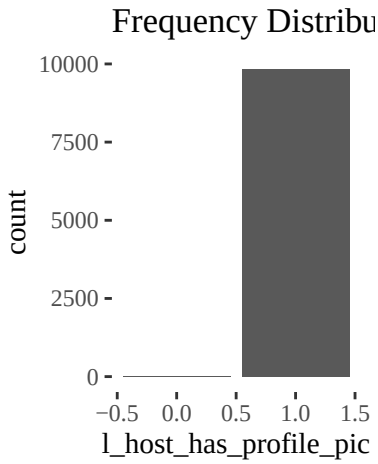
| Var.1 | RMSE | Mean.price | RMSE.price |
|---|---|---|---|
| Apartment size | | | |
| 2 | 27.7 | 42 | 0.66 |
| 3 | 25.5 | 44 | 0.58 |
| 4 | 42.8 | 60 | 0.71 |
| 5 | 43.2 | 72.3 | 0.6 |
| 6 | 40.6 | 85.2 | 0.48 |
| Type | | | |
| apartment | 26.8 | 48.3 | 0.55 |
| condominium | 33.2 | 46.4 | 0.72 |
| Neighbourhood | | | |
| Khlong Toei | 24.9 | 43.8 | 0.57 |
| Vadhana | 35 | 63.3 | 0.55 |
| Huai Khwang | 24.2 | 37.7 | 0.64 |
| Ratchathewi | 39 | 54.9 | 0.71 |
| Sathon | 21.8 | 48.1 | 0.45 |
| All | 32.6 | 49.6 | 0.66 |

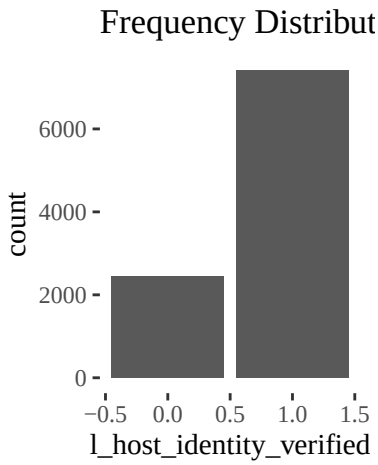


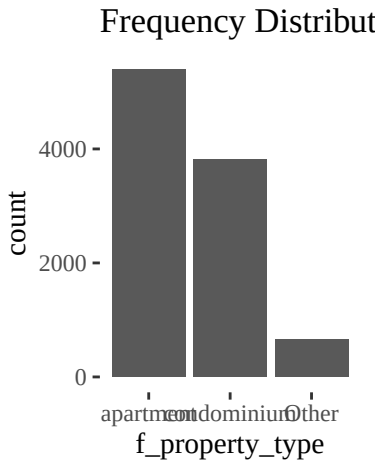

Frequency Distribut
6000
4000
2000
0
count
Din Daeng/Huai Khwang
Khlong Toei
Sukhumvit
f_host_neighbourhood

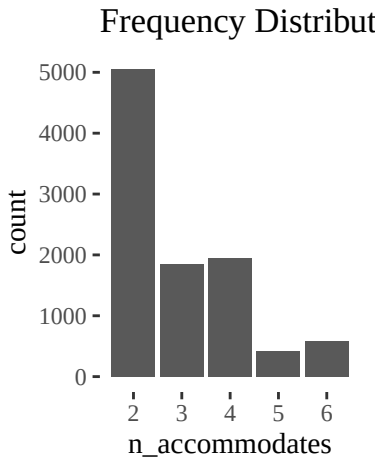

Frequency Distribut
2000
1500
1000
500
0
count
0
250
500
750
1000
n_host_listings_count

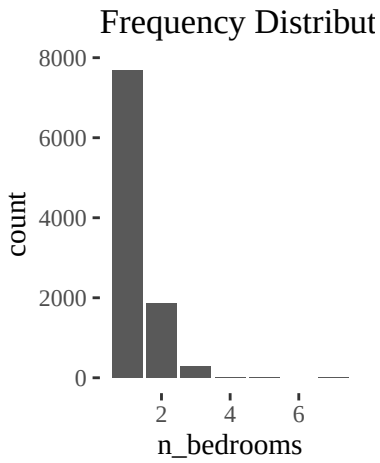

Frequency Distribut
2500
2000
1500
1000
500
0
count
0
5
10
n_host_verifications

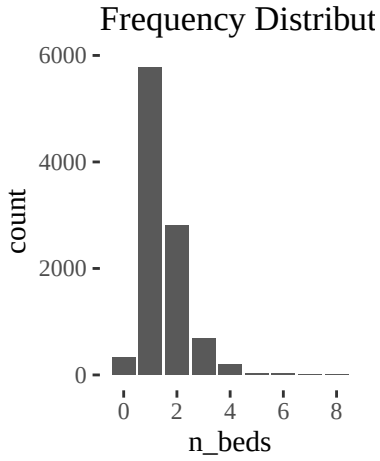

Frequency Distribu
10000
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
l_host_has_profile_pic

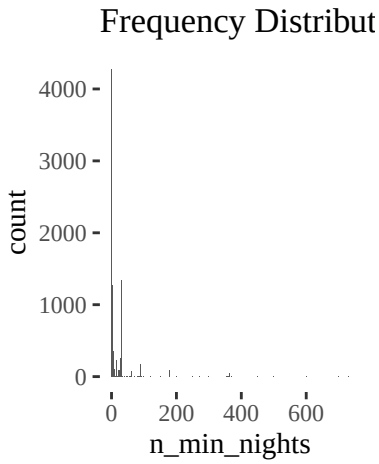

Frequency Distribut
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
l_host_identity_verified

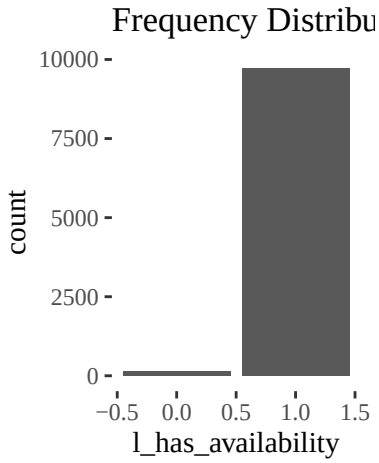

Frequency Distribut
4000
2000
0
count
apartment
condominium
Other
f_property_type

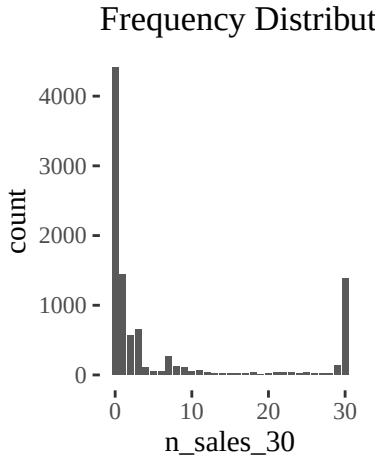

Frequency Distribut
5000
4000
3000
2000
1000
0
count
2
3
4
5
6
n_accommodates

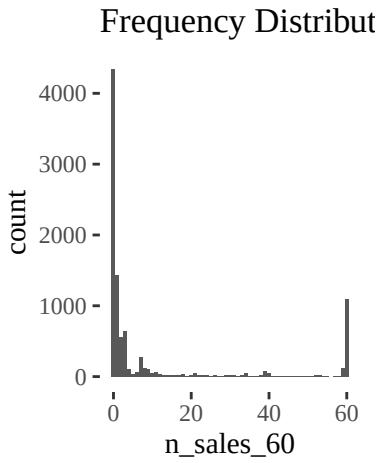

Frequency Distribut
8000
6000
4000
2000
0
count
2
4
6
n_bedrooms

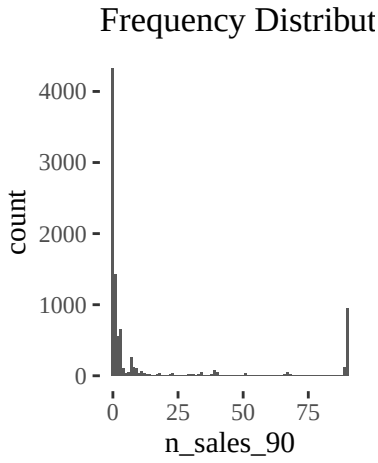

Frequency Distribut
6000
4000
2000
0
count
0
2
4
6
8
n_beds

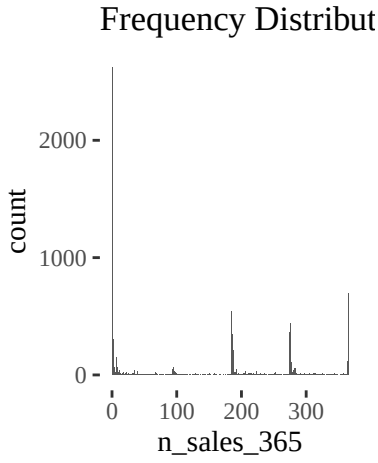

Frequency Distribut
4000
3000
2000
1000
0
count
0
200
400
600
n_min_nights

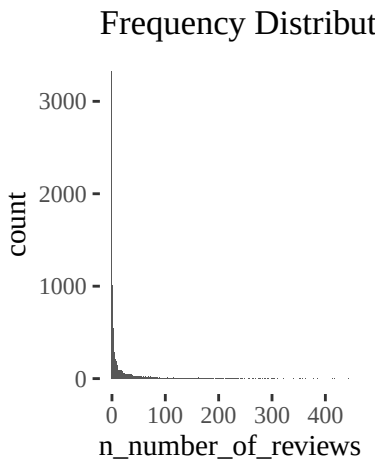

Frequency Distribu
10000
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
l_has_availability

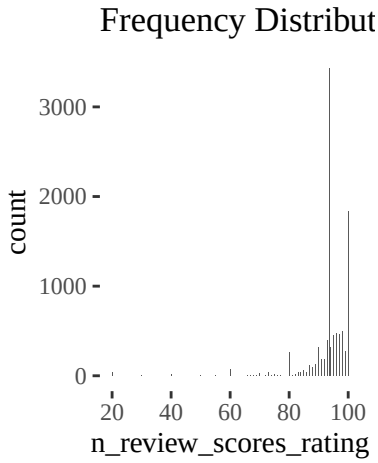

Frequency Distribut
4000
3000
2000
1000
0
count
0
10
20
30
n_sales_30

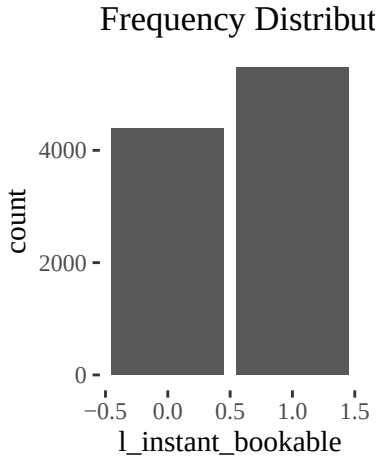

Frequency Distribut
4000
3000
2000
1000
0
count
0
20
40
60
n_sales_60

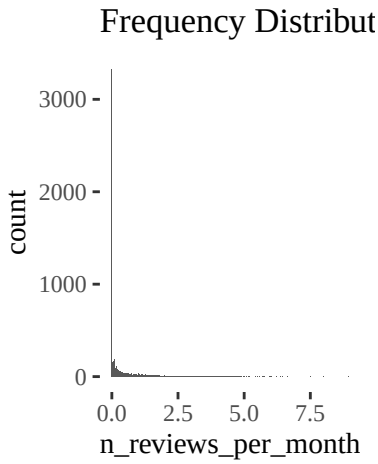

Frequency Distribut
4000
3000
2000
1000
0
count
0
25
50
75
n_sales_90

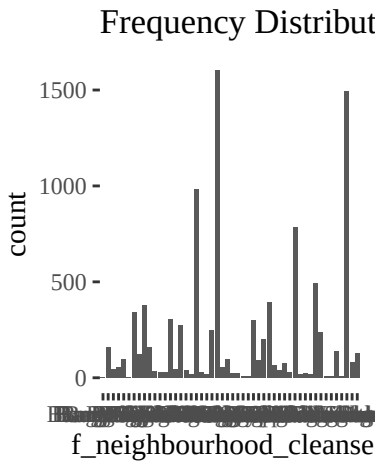

Frequency Distribut
2000
1000
0
count
0
100
200
300
n_sales_365

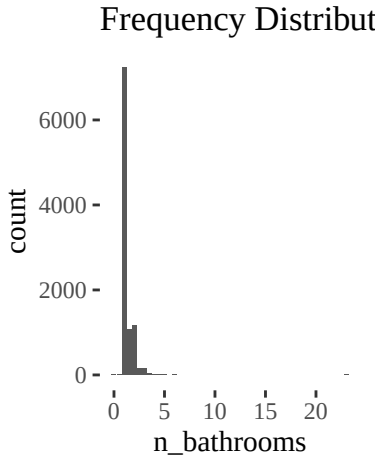

Frequency Distribut
3000
2000
1000
0
count
0
100
200
300
400
n_number_of_reviews

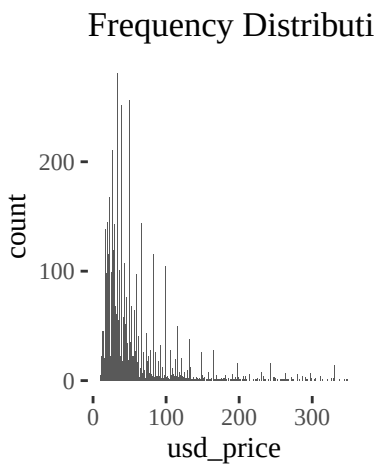

Frequency Distribut
3000
2000
1000
0
count
20
40
60
80
100
n_review_scores_rating

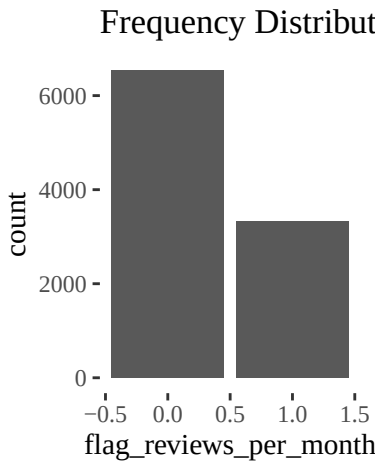

Frequency Distribut
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
l_instant_bookable

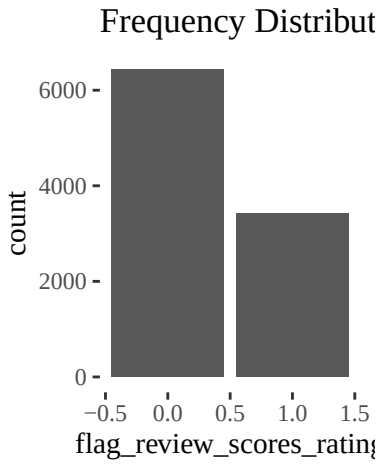

Frequency Distribut
3000
2000
1000
0
count
0.0
2.5
5.0
7.5
n_reviews_per_month

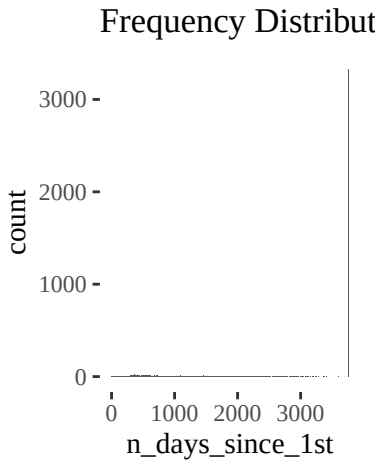

Frequency Distribut
1500
1000
500
0
count
f_neighbourhood_cleanse

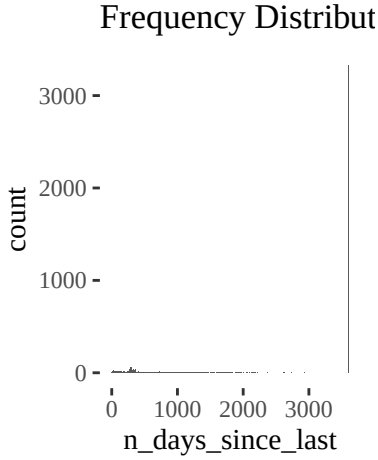

Frequency Distribut
6000
4000
2000
0
count
0
5
10
15
20
n_bathrooms

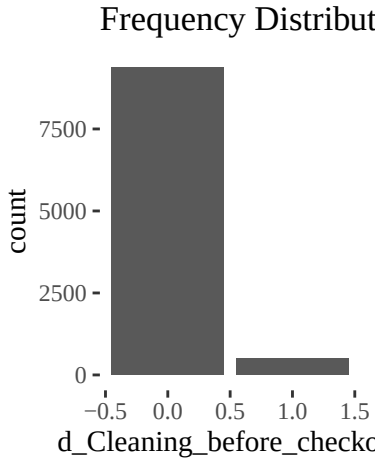

Frequency Distributi
count
200
100
0
0
100
200
300
usd_price

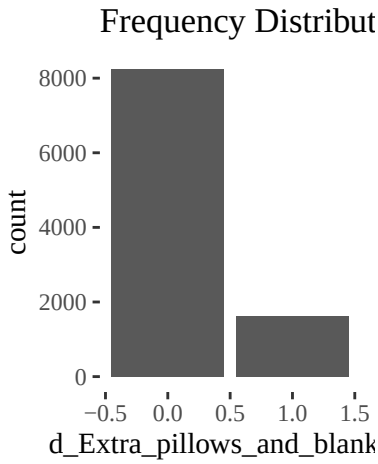

Frequency Distribut
count
6000
4000
2000
0
−0.5
0.0
0.5
1.0
1.5
flag_reviews_per_month

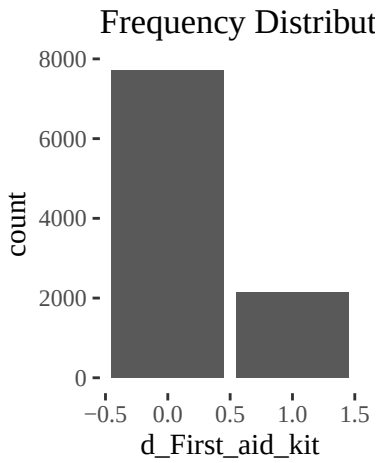

Frequency Distribut
count
6000
4000
2000
0
−0.5
0.0
0.5
1.0
1.5
flag_review_scores_ratin

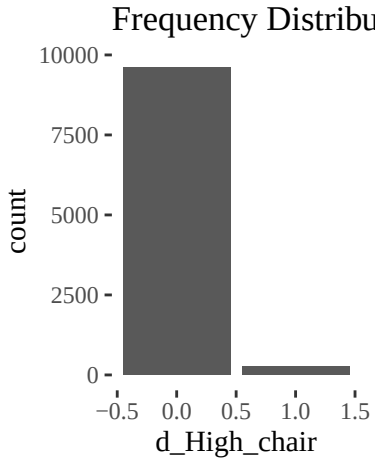

Frequency Distribut
3000
2000
1000
0
count
0
1000
2000
3000
n_days_since_1st

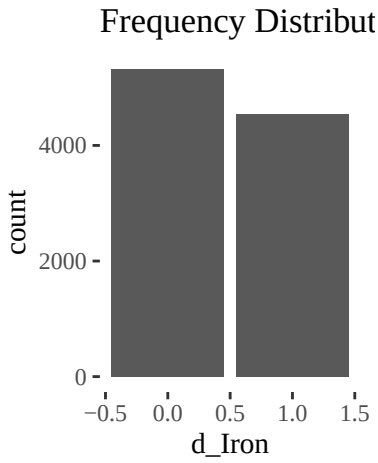

Frequency Distribut
3000
2000
1000
0
count
0
1000
2000
3000
n_days_since_last

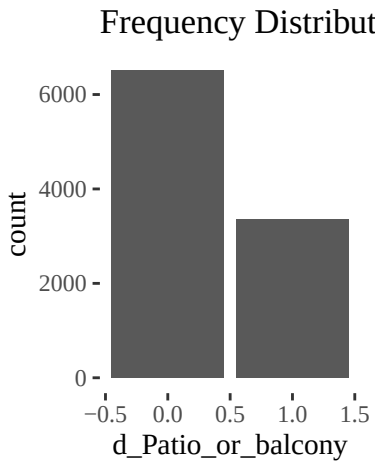

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Cleaning_before_checkc

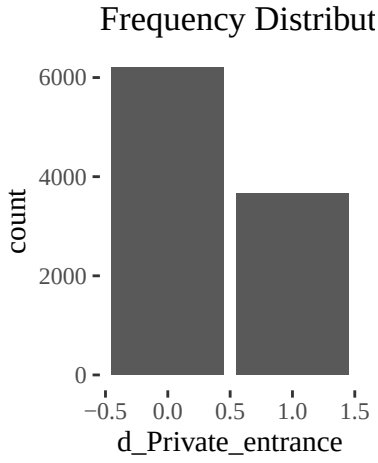

Frequency Distribut
8000
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Extra_pillows_and_blank

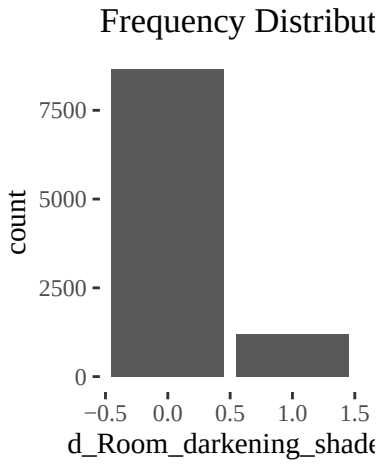

Frequency Distribut
8000
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_First_aid_kit

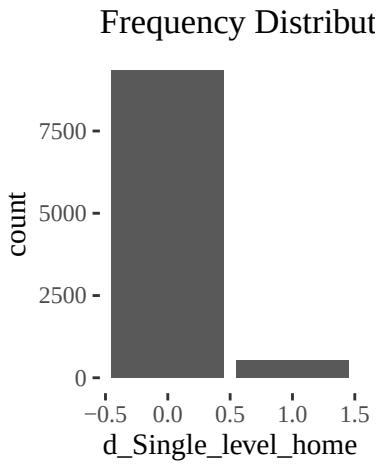

Frequency Distribu
10000
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_High_chair

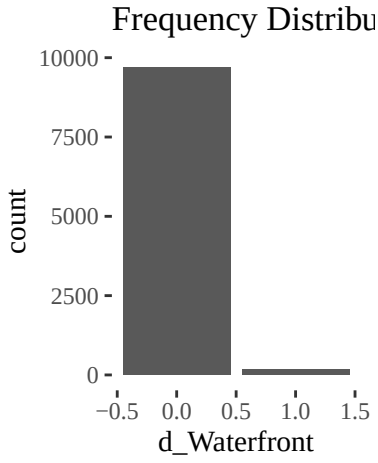

Frequency Distribut
count
4000
2000
0
−0.5
0.0
0.5
1.0
1.5
d_Iron

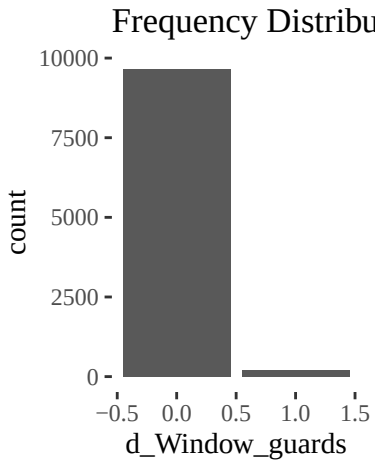

Frequency Distribut
count
6000
4000
2000
0
−0.5
0.0
0.5
1.0
1.5
d_Patio_or_balcony

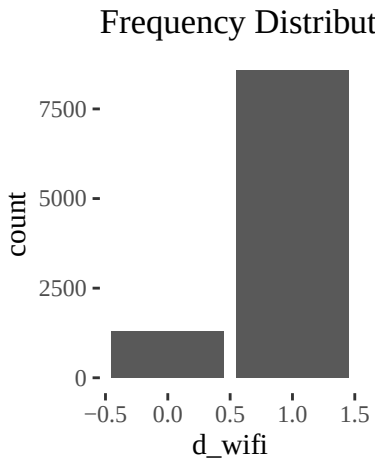

Frequency Distribut
count
6000
4000
2000
0
−0.5
0.0
0.5
1.0
1.5
d_Private_entrance

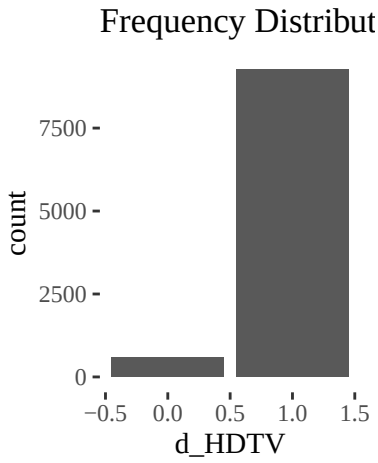

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Room_darkening_shade

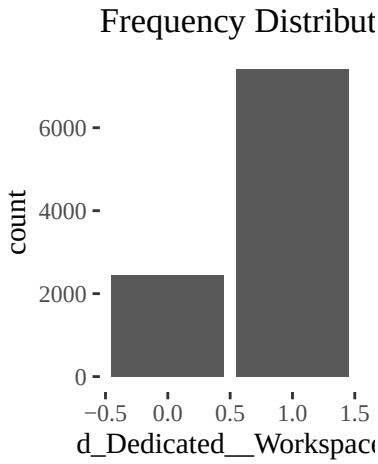

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Single_level_home

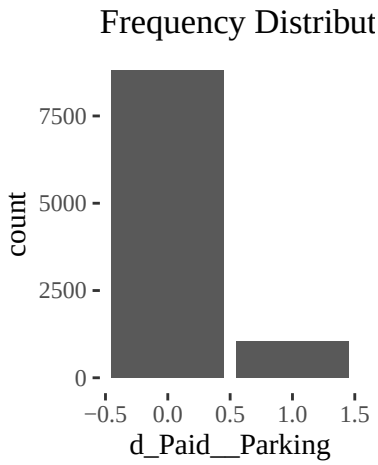

Frequency Distribu
10000
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Waterfront

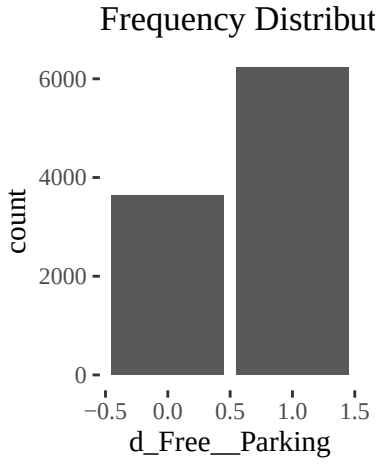

Frequency Distribu
10000
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Window_guards

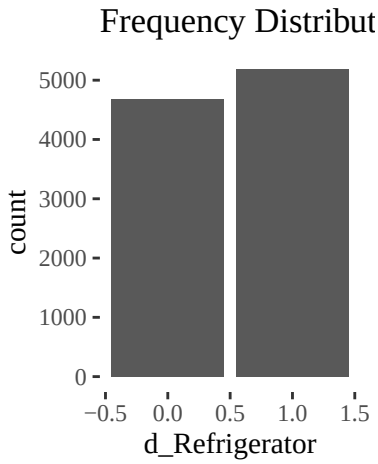

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_wifi

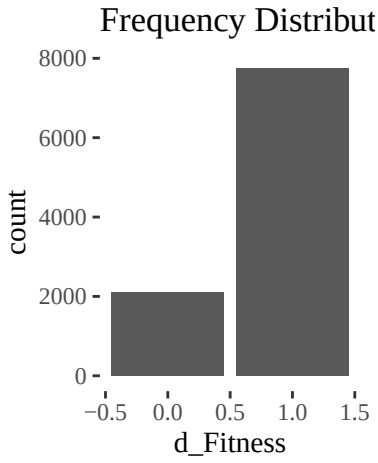

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_HDTV

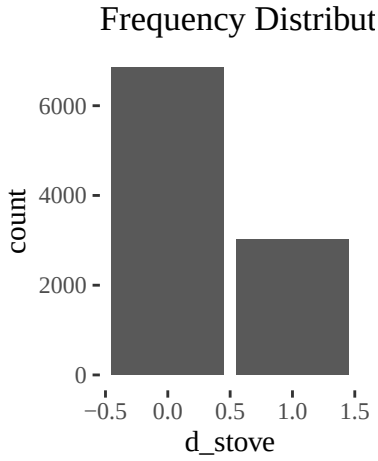

Frequency Distribut
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Dedicated__Workspac

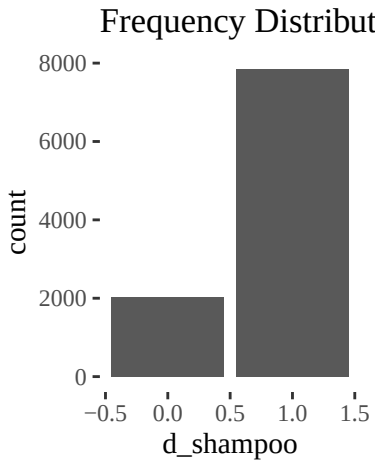

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Paid__Parking

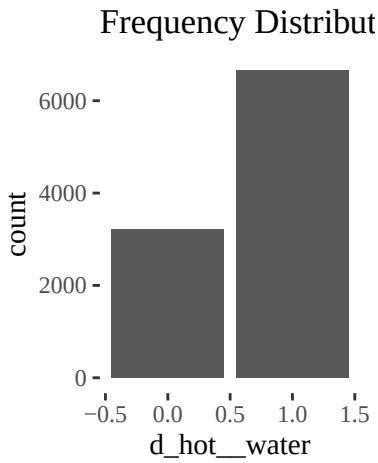

Frequency Distribut
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Free__Parking

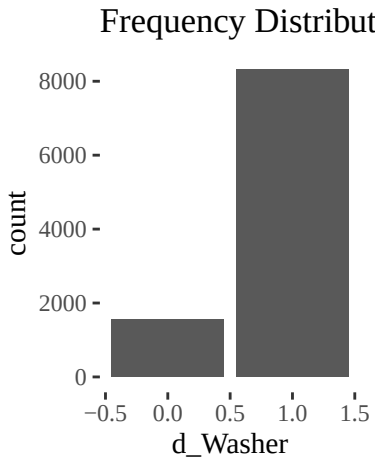

Frequency Distribut
5000
4000
3000
2000
1000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Refrigerator

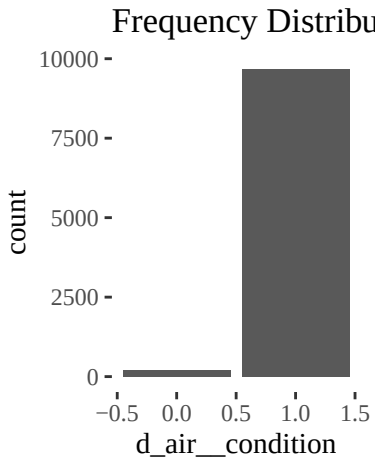

Frequency Distribut
8000
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Fitness

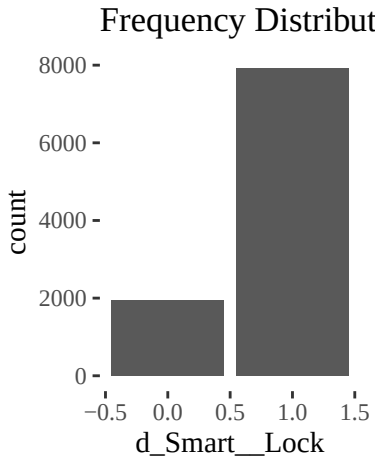

Frequency Distribut
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_stove

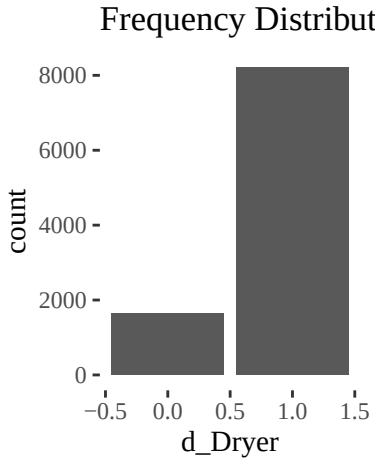

Frequency Distribut
8000
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_shampoo

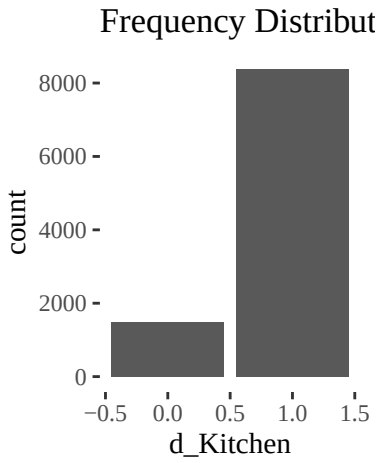

Frequency Distribut
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_hot__water

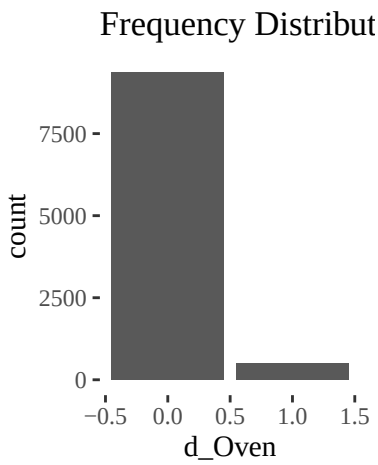

Frequency Distribut
8000
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Washer

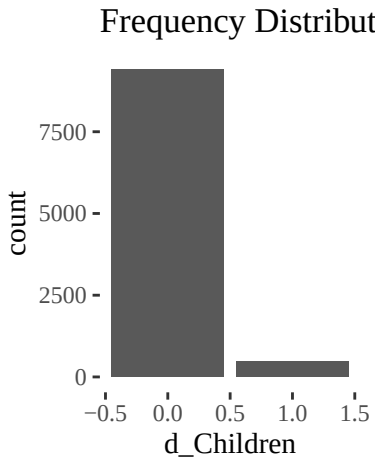

Frequency Distribu
10000
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_air__condition

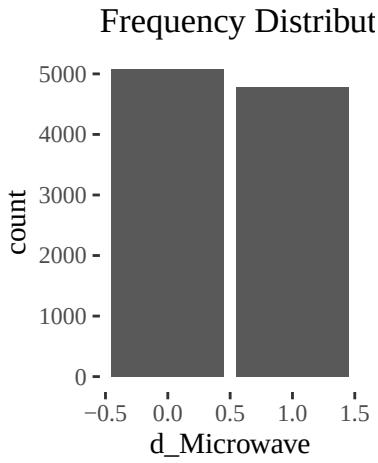

Frequency Distribut
8000
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Smart__Lock

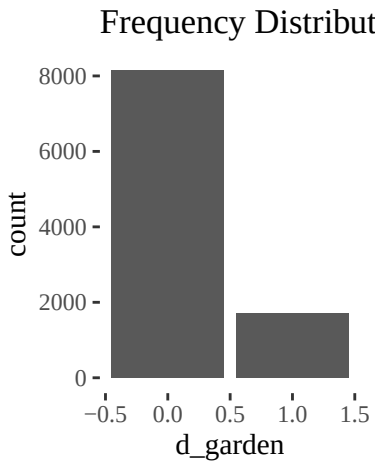

Frequency Distribut
8000
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Dryer

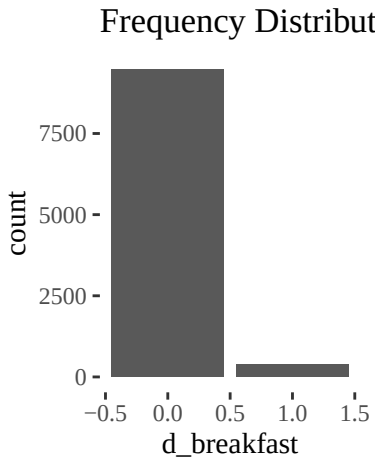

Frequency Distribut
8000
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Kitchen

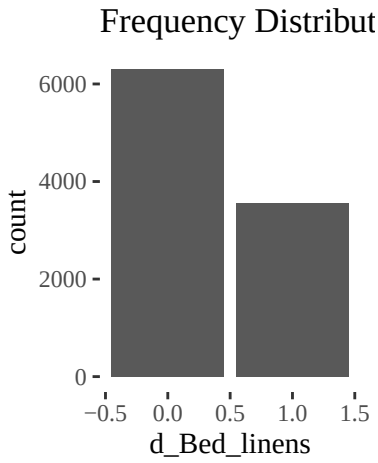

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Oven

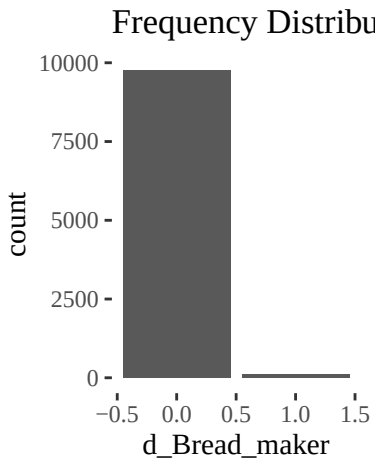

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Children

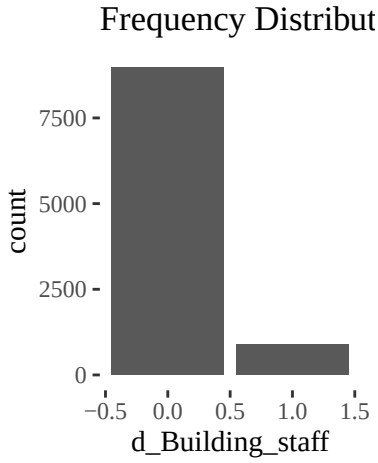

Frequency Distribut
5000
4000
3000
2000
1000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Microwave

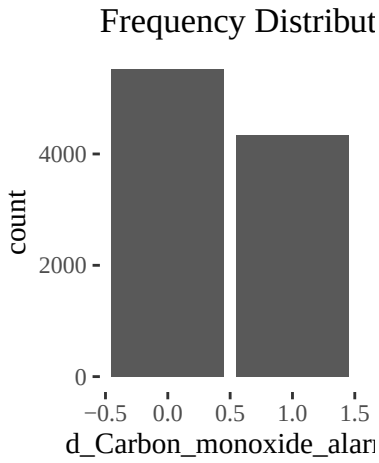

Frequency Distribut
8000
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_garden

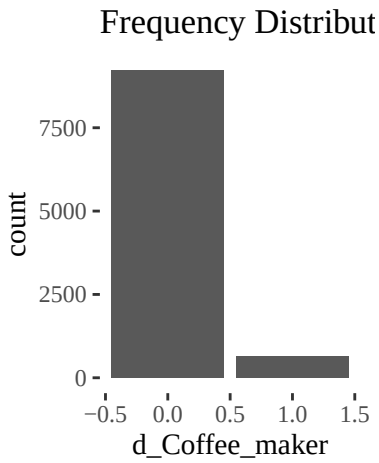

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_breakfast

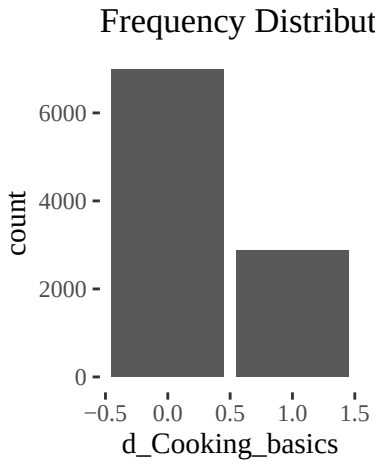

Frequency Distribut
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Bed_linens

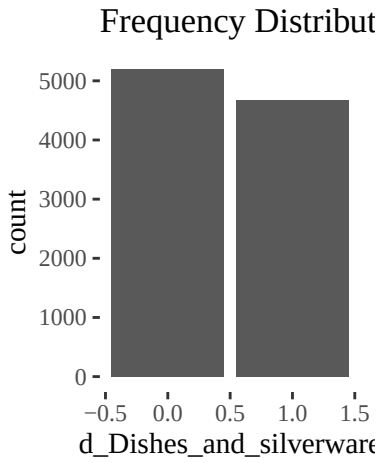

Frequency Distribu
10000
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Bread_maker

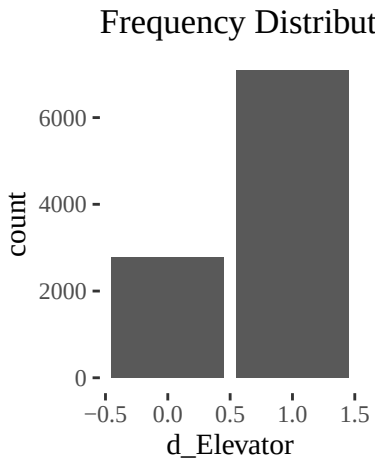

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Building_staff

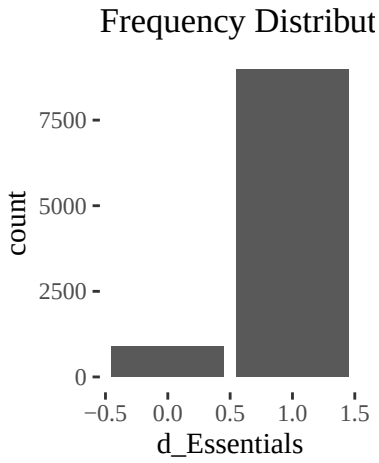

Frequency Distribut
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Carbon_monoxide_alar

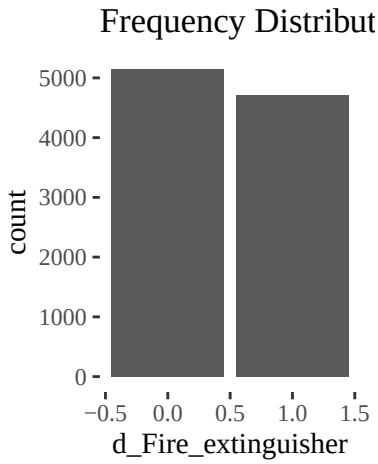

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Coffee_maker

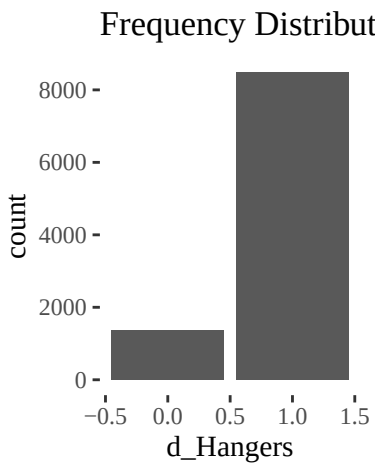

Frequency Distribut
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Cooking_basics

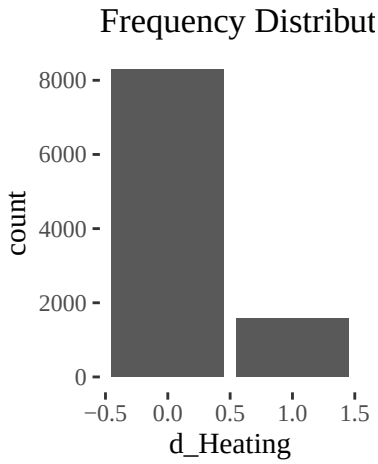

Frequency Distribut
5000
4000
3000
2000
1000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Dishes_and_silverware

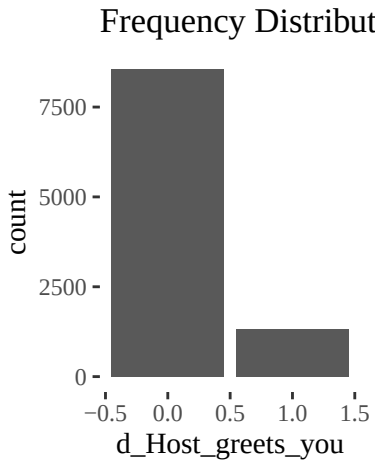

Frequency Distribut
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Elevator

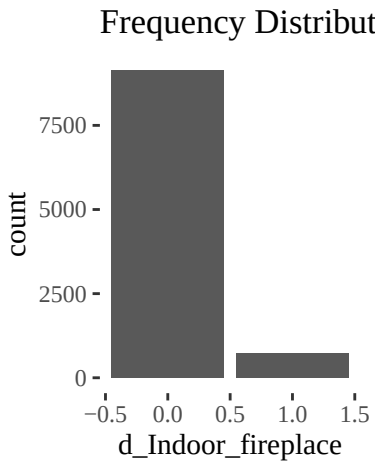

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Essentials

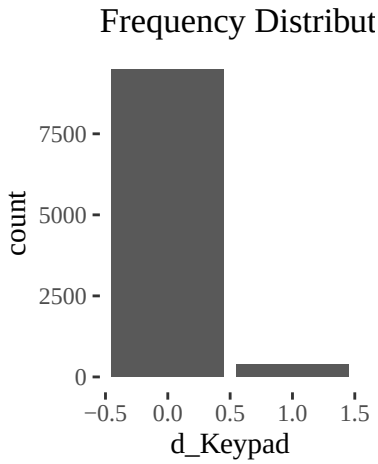

Frequency Distribut
5000
4000
3000
2000
1000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Fire_extinguisher

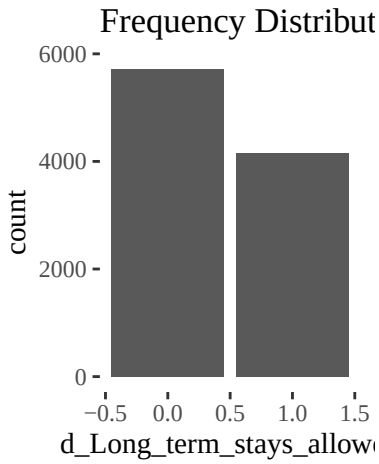

Frequency Distribut
8000
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Hangers

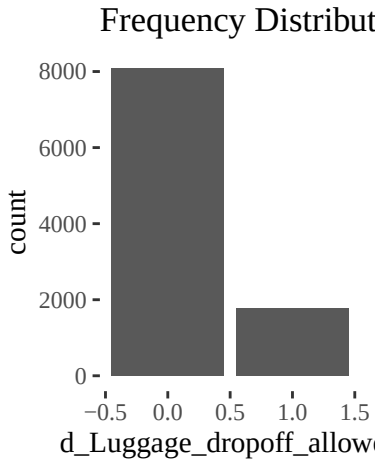

Frequency Distribut
8000
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Heating

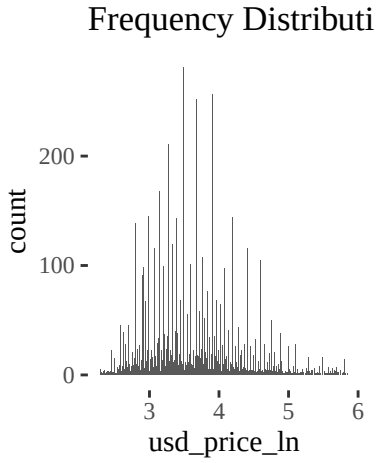

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Host_greets_you

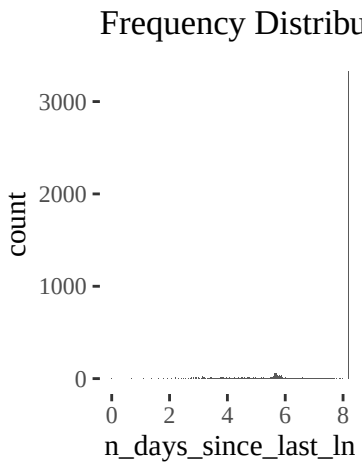

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Indoor_fireplace

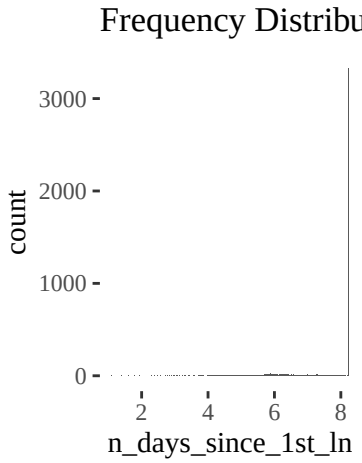

Frequency Distribut
7500
5000
2500
0
count
−0.5
0.0
0.5
1.0
1.5
d_Keypad

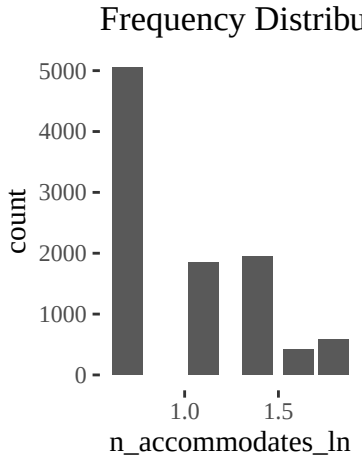

Frequency Distribut
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Long_term_stays_allow

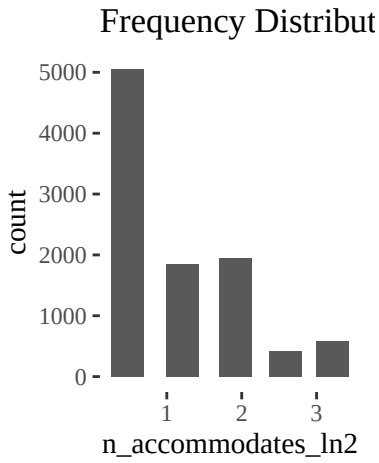

Frequency Distribut
8000
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
d_Luggage_dropoff_allow

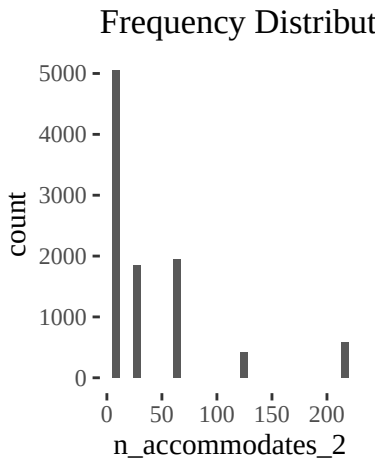

Frequency Distributi
200
100
0
count
3
4
5
6
usd_price_ln

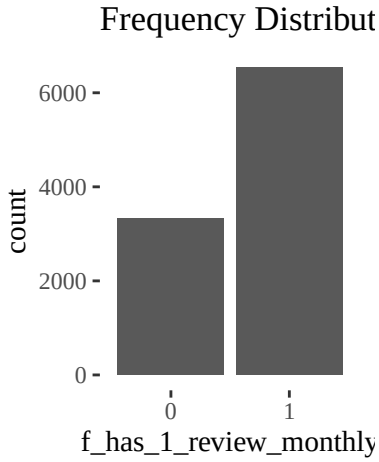

Frequency Distribut
3000
2000
1000
0
count
0
2
4
6
8
n_days_since_last_ln

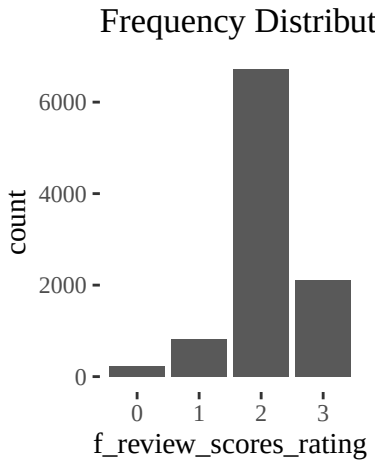

Frequency Distribut
3000
2000
1000
0
count
2
4
6
8
n_days_since_1st_ln

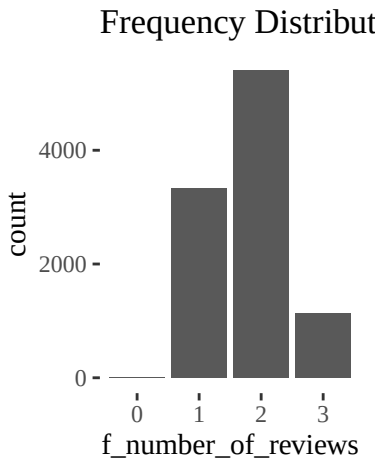

Frequency Distribut
5000
4000
3000
2000
1000
0
count
1.0
1.5
n_accommodates_ln

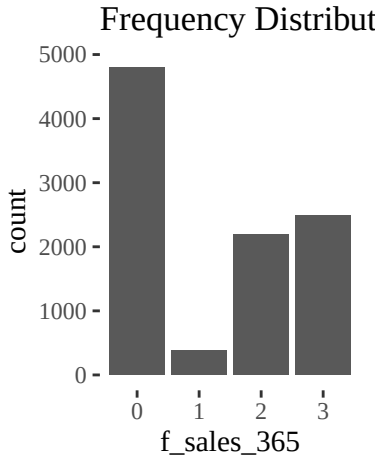

Frequency Distribut
5000
4000
3000
2000
1000
0
count
1
2
3
n_accommodates_ln2

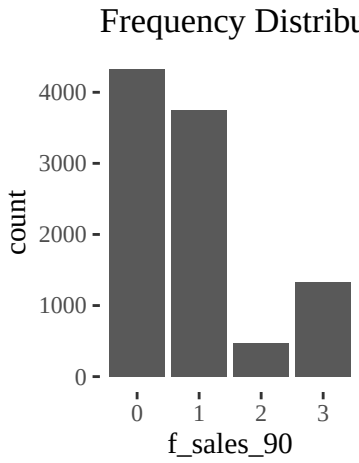

Frequency Distribut
5000
4000
3000
2000
1000
0
count
0
50
100
150
200
n_accommodates_2

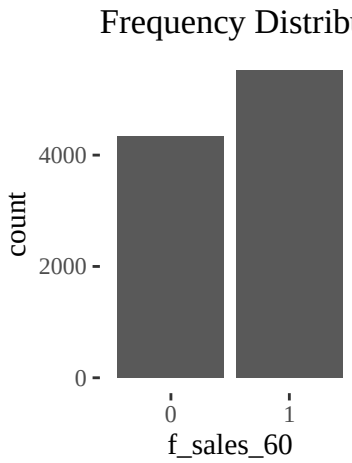

Frequency Distribut
6000
4000
2000
0
count
0
1
f_has_1_review_monthly

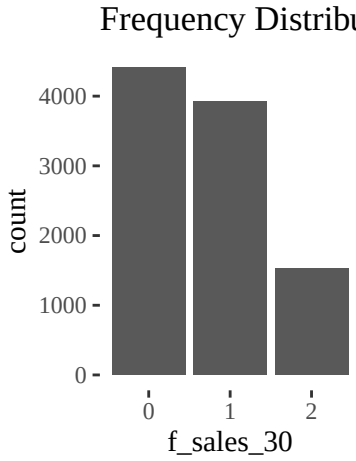

Frequency Distribut
6000
4000
2000
0
count
0
1
2
3
f_review_scores_rating

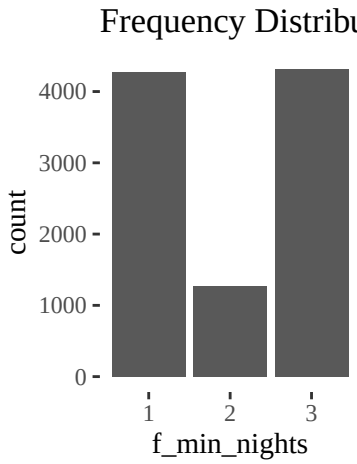

Frequency Distribut
4000
2000
0
count
0
1
2
3
f_number_of_reviews

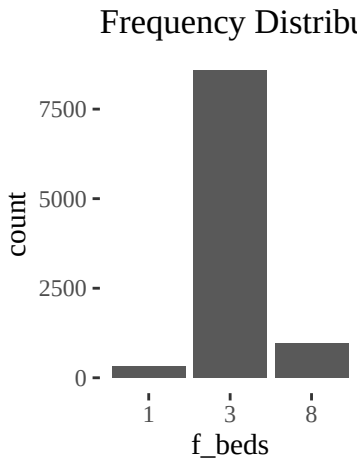

Frequency Distribut
5000
4000
3000
2000
1000
0
count
0
1
2
3
f_sales_365

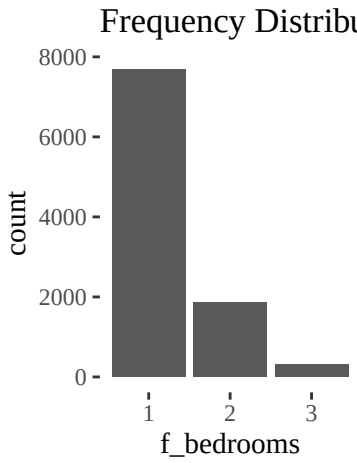

Frequency Distribut
4000
3000
2000
1000
0
count
0
1
2
3
f_sales_90

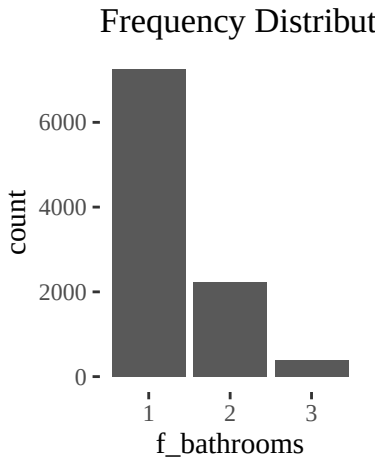

Frequency Distribut
4000
2000
0
count
0
1
f_sales_60

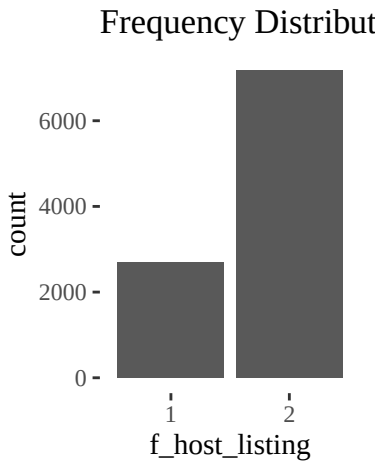

Frequency Distribut
4000
3000
2000
1000
0
count
0
1
2
f_sales_30

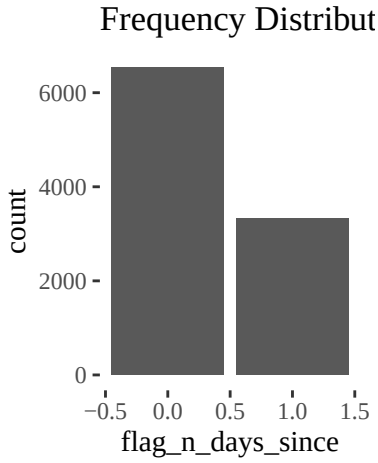

Frequency Distribut
4000
3000
2000
1000
0
count
1
2
3
f_min_nights

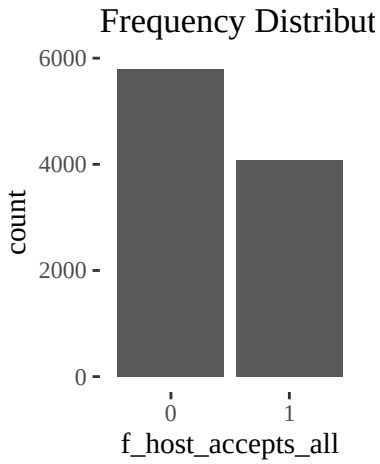

Frequency Distribut
7500
5000
2500
0
count
1
3
8
f_beds


Frequency Distribut
8000
6000
4000
2000
0
count
1
2
3
f_bedrooms


Frequency Distribut
6000
4000
2000
0
count
1
2
3
f_bathrooms


Frequency Distribut
6000
4000
2000
0
count
1
2
f_host_listing


Frequency Distribut
6000
4000
2000
0
count
−0.5
0.0
0.5
1.0
1.5
flag_n_days_since


Frequency Distribut
6000
4000
2000
0
count
0
1
f_host_accepts_all

NULL