All the cleaning, imputation, transformation and feature engineering stated below for the Analysis Data, have been applied to the Scoring Data as well.

Introduction:

The pivotal motive for the exercise is to predict rental prices for Airbnb listings, using a given analysis data set, containing 36839 rows of data, and 91 columns or variables including price. The goal was to build the best price predicting model, whilst not overfitting, and the final assessment for our model would be done by the *rmse* of the predicted prices, as applied on the scoring data set.

Exploratory Data Analysis:

The very first thing I did was to eye-ball all of the 90 variables, understand their distribution, the type of information they revealed, and try to use **intuitive judgement** to understand which of the given variables might be very important, which might be totally useless, and most importantly, which of the variables needed feature engineering of some kind, to be made useful.

I discovered there are **3 types of NA's** in the data; some columns such as ID, Host Listings Count, and others need to be **dropped immediately**; some columns such as State and Market have **similar information**; and four columns relating to maximum nights had **outliers**.

Then, I made a correlation chart *(Appendix I)* of the numeric variables, and discovered that accommodates and beds show a strong positive correlation of 0.78, followed by accommodates and bedrooms, and some others, which might play a significant role in selecting the final variables to use.

Data Cleaning:

I started off by **dropping** those variables from my dataset, which I believed would not contribute towards a good model, some of them being Host Listings Count, State, Country, License, etc.

Price: I made a histogram of Price distribution *(Appendix II)*, and discovered there were around 15 rows with astronomical values, Price = 0, and needed to be removed for calculations.

Zipcode: Some zipcodes were mis-represented, they were converted to the correct format.

Removing Outliers:

Outliers were noticed in **4 columns**: Maximum Nights, Minimum Maximum Nights, Maximum Maximum Nights, and Maximum Nights Avg Ntm; contained values equal to 2.15 billion nights. I made scatter plots *(Appendix III)*, and discovered that were very few points above 1200 nights. So, I capped the maximum value to 1200 nights, and imputed all values above 1200 with 1200.

Missing Values:

Firstly, I made a list of the **percentage of NA's** in all the columns. Columns with high percentage were dropped: Host Acceptance Rate (100%), Square Feet (99%), Weekly Price (90%), Monthly Price (99%).

Text: For Summary, Space, Description and others, median of the word count was used.

Logical value: For Host is Superhost and others, missing values were filled with ' f ' (or False).

Host Response Time: The NA's here was replaced by the maximum value of ' a few days or more '.

Numeric: Median was used to replace NA's for Host Response Rate, Security Deposit, and others.

Dates: For Host Since and others, after feature engineering, median days was used to fill NA's.

Zipcodes: For imputing missing Zipcodes, the mode of Zipcode was used.

For all imputations, mean was used initially, but **Median** gave a better result. Also, **Mice Package** was used to impute all the above-stated missing values, but no significant difference was observed.

Data Transformation:

Text: For Name, Summary, Space, Notes, Transit, Access, Host About and others, I did the word count of the number of words in each value, using the space separator " ".

Logical Value: For Host is Superhost, Is Location Exact and others, they were revalued by replacing ' t ' as TRUE and ' f ' as FALSE, and further converted to logical format, for better interpretation.

Host Response Rate: Since the values were in factor form, they were converted to strings, and the percentage sign was removed. Further, it was converted to numeric format.

Feature Engineering:

Keyword: The very first feature engineering I did was to count the keywords: 'Luxury', 'luxury', 'Luxurious', and 'luxurious', in all the text columns, or simply concatenate the text columns and count for 'luxur', ignoring the case. This indeed turned out to be an important new feature for the models.

Uppercase: I made a new feature 'name_upper', wherein I counted the uppercase letters in Name.

List Format: Host Verifications and Amenities were converted to list format, wherein I counted the number of commas, and further added 1 to the count, as the last value in each of them didn't have a comma at the end, and also, say if there was only 1 amenity, it would otherwise get counted as 0.

Amenities: Amenities was further divided into different logicals, to understand if a particular row had that specific amenity or not. New columns created were: Wifi, Internet, TV, Parking, and others.

Date Columns: The number of days was counted, from the date mentioned, till the System Date.

Analysis and Model Building:

After my data was clean and transformed, I converted them to a new .csv file to save the effort of repeating the procedure. The first step was to split the data to train and test using the Create Data Partition function, I split the data 80:20. All models were built on train, and then on the full data.

Initially, without exploring feature selection techniques, I started my quest by considering all variables, to build all the possible models such as Linear, Logistic, Lasso, Ridge, Stepwise, Trees, all the way to Bagging and Gradient Boosting. This gave me a clear idea as to which models would be suitable for my data, and were returning the lowest rmse's. I tried re-running all the models for a few times by tweaking the input parameters, and also, I started small, by not putting greater than 300 trees for the tree models. I gained insight that bootstrapping models would be the best approach.

So, I immediately diverted my focus to building and refining the bootstrapping models, Random Forest and GBM to be precise. Using Random Forest and GBM, I could get the rmse to drop to 62-63, but I was not being able to reduce it further. I switched to the GBM model, and started tweaking the parameters further. At this stage, I had not considered feature selection. I increased the trees to 50,000, which is quite high, and got the rmse of 60.16. Any further modifications were not working.

Then I felt the need to do feature selection, and ran Hybrid Stepwise, and a Combination of Lasso and Ridge to check for important variables. Even after feature selection, the rmse didn't seem to drop. It is now, that I went back to my original data cleaning file, and started re-working on it. At this stage, I grouped levels and categories for some variables, such as Property type, and converted beds to Real Bed? Yes/No format, but I failed to improve the rmse.

Finally, I created some new columns, such as keyword search, uppercase in name, and average minimum nights, and average maximum nights; and also inserted new columns which I had ignored earlier, such as zipcode and calculated host listings counts (4 columns). I ran the models again, this time, with an improvement in the rmse, I scored 57-58 on my models.

Going forward, after's my professor's advice, I tried the Ranger and the XGBOOST models. I realized these models were extremely fast compared to any other model I tried, and also gave a lower train/test rmse. The XGBOOST model was what yielded my best submission rmse score of 56.3614 *(Appendix IV)*. Initially, I used the model with default parameters, but since this model was at least 10x faster, I played around a lot with the parameters. I often ended up overfitting my data, but trial and error worked for me, and gave me an acceptable result.

Long story short, the competition was extremely beneficial for personal learning, and to re-define my competencies in R. It facilitated good hands on learning, different from what one could learn in class. The project was challenging in every way, compelling the participants to keep trying harder, even though we might have scored an acceptable rmse. I am appreciative of the fact that I explored many different models, each one taking the learning a step ahead.

I made a total of 35 submissions, and after almost every submission, I felt like I had the best model in place, but didn't work. This forced me to further clean and transform my data, and create new features which might be essential for better prediction.

Personal Insights after completing the project:

Making the transition all the way from a pure business background to the world of data science was a daunting challenge, yet an interesting one. I realized my approach to almost everything I did initially was way out of a Data Scientist's way, and I was still thinking from an unfitting mindset.

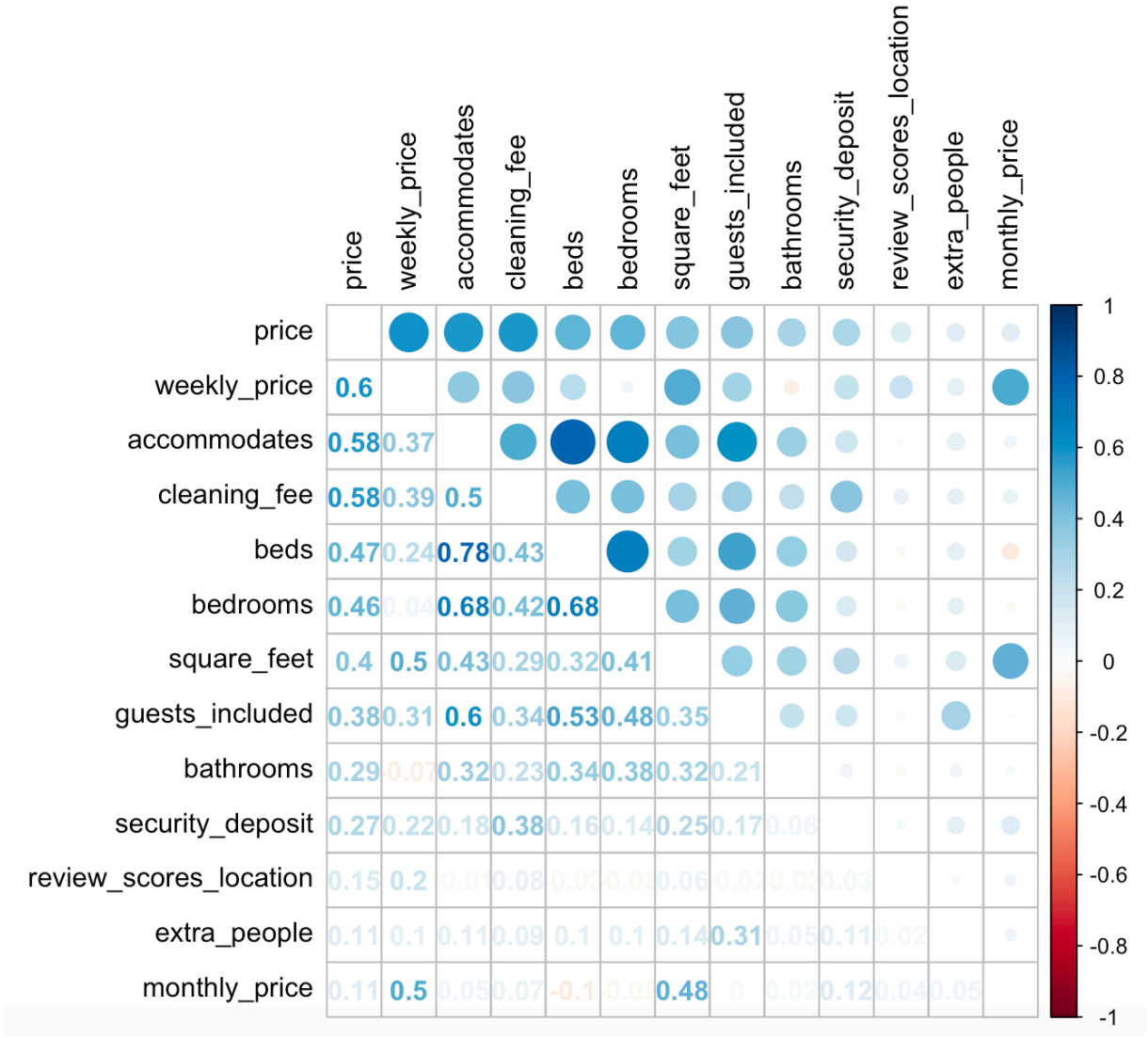
My initial days were spent on unrealistic models, wherein I was either considering all the variables at once, or only few of them, but without doing any feature selection. Though some models like Linear Regression and Trees gave results back in a few minutes, but were mediocre and unacceptable. When I shifted to GBM model, they used to run for 3-4 hours at least, and yet yielded unsatisfactory results.

Only after I realized that the data needed a lot more cleaning, a lot more transformation, and a lot more feature engineering, was I able to rise up the leaderboard. Now, I was thinking like a Data Scientist, and could focus more my data, rather than only the models, and results.

I realized only later that, I could have the best results, only when my data was good, and simultaneously I was using efficient advanced models with tuned parameters.

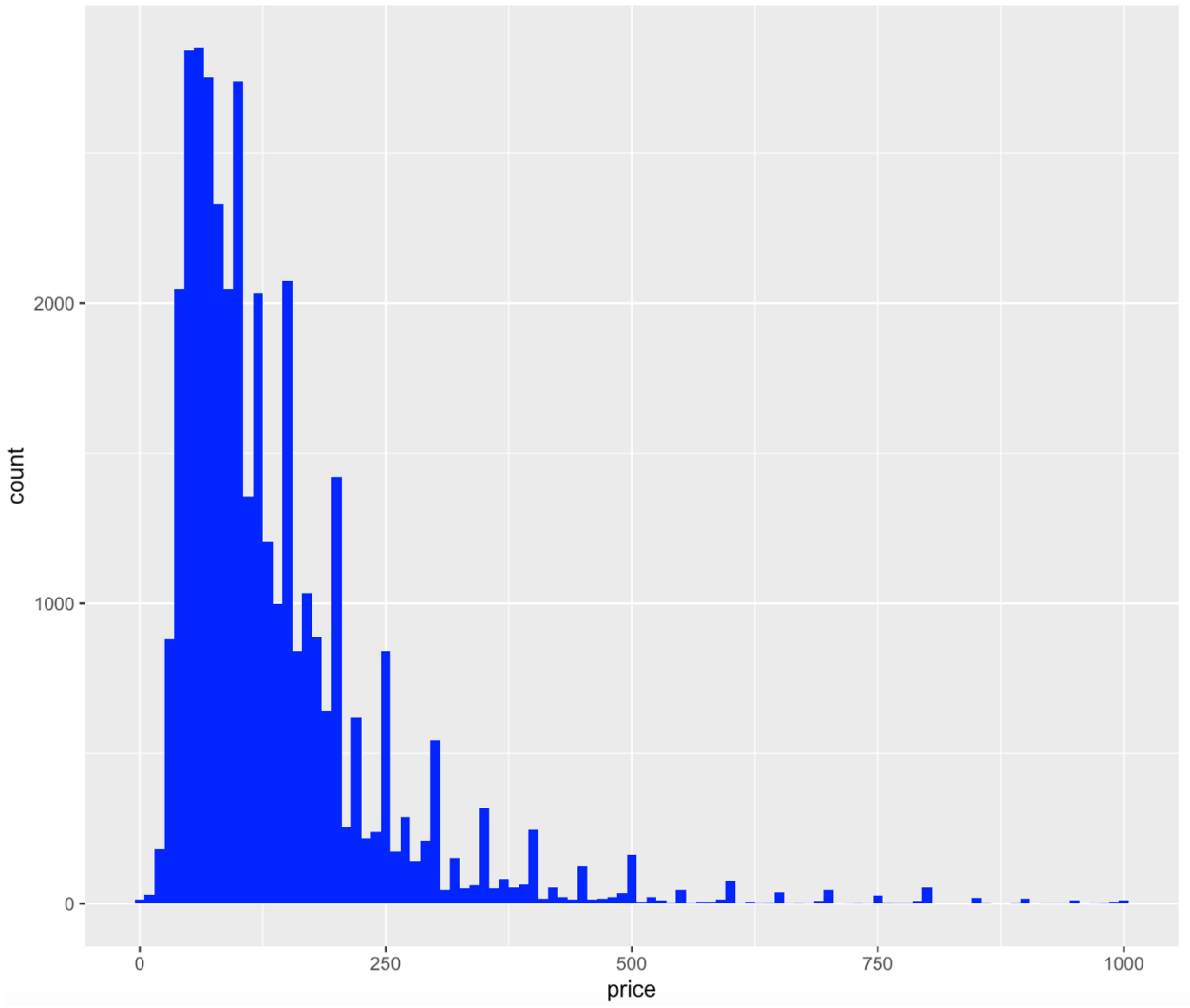
Now I know, to a large extent, what not to waste time on!

Appendix I: **Correlation chart of Numeric Variables**



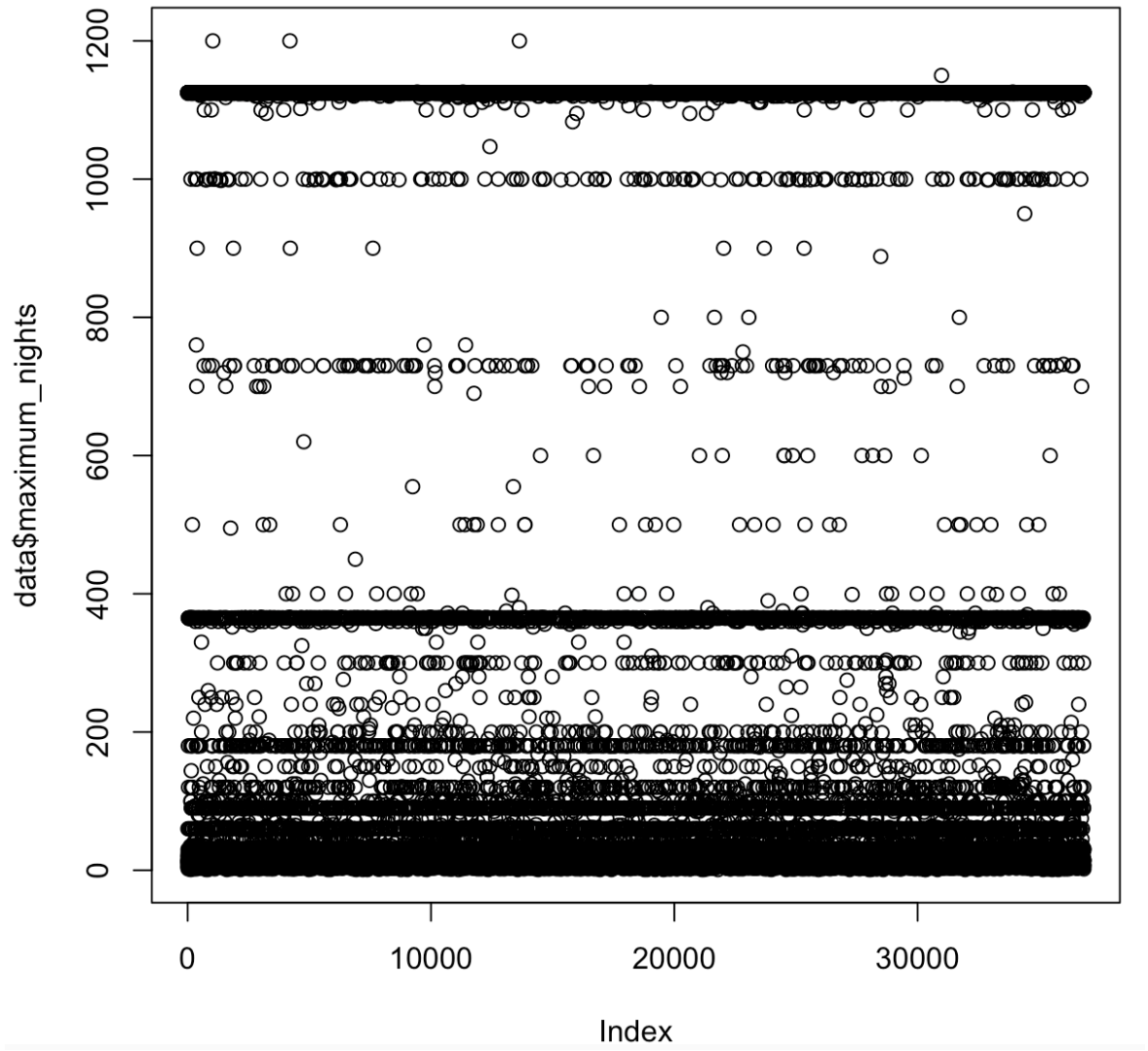
Accommodates and Beds show a strong positive correlation

Appendix II: **Histogram of Price Distribution**



15 rows had Price = $0

Appendix III: **Distribution of Maximum Nights variable**



There were only few values above 1200 nights

Appendix IV: **R Code**

Please see attached .R script file.